XOR

S-100-12
COMPUTER ASSEMBLY
AND
OPERATING INSTRUCTIONS

FORWARD

This manual is divided into two chapters for your convenience.

Chapter One, ASSEMBLY, contains the initial unpacking, assembly, and testing instructions.

Chapter Two, GENERAL OPERATING PROCEDURES, consists of technical information on systems hardware and software. It includes power up, systems utilities, and other information to get your system up and running correctly.

**********TABLE OF CONTENTS**********

CHAPTER ONE - ASSEMBLY	PAGE
1.0 UNPACKING	3
1.1 INSPECTION	3
1.2 CABINET ASSEMBLY	3
1.3 POWER TRANSFORMER CONNECTIONS	3
1.4 DISK DRIVE OPTIONS	4
1.5 DISK DRIVE MOUNTING	4
1.6 INITIAL POWER ON	4
1.7 BOARD INSTALLATION	5
1.8 SERIAL TERMINAL CONNECTION	5
1.9 COMPUTER TERMINAL CHECKS	5
1.10 DISK OPERATION CHECKS	5
1.11 FINAL CLOSE OUT	6
1.12 DRIVE NOTES	6

CHAPTER TWO - GENERAL OPERATING PROCEDURES	
2.1 SHOCK AND TILT SENSORS	7
2.2 HARD DISK SYSTEMS	8
2.3 POWER UP	8
2.4 MONITOR COMMANDS	9
2.5 DISK BOOT COMMANDS	10
2.6 BOOTING THE OPERATING SYSTEM	10
2.7 BACKING UP	12
2.8 SAFETY PRECAUTIONS	14
2.9 SERVICE POLICY	15
2.10 DISK A - XOR DISK UTILITIES	16
2.11 DISK B - SYSTEM SOURCE DISK	20
2.12 MODIFYING AND GENERATING CP/M OPERATING SYSTEMS	24
2.13 DSYSGEN OPERATING INSTRUCTIONS	28
2.14 DSYSGEN STEP-BY-STEP	32
2.15 MP/M II SYSTEM GENERATION PROCEDURE	34
2.16 MEANING OF BITS IN CODE BYTE	41
2.17 SUB TYPE FILES FOR SYSTEM BUILDING	43
2.18 MEMORY MAP	46
2.19 APPENDICES	
PRINTER DRIVERS	
CENTRONICS INTERFACE	47
DIABLO SERIAL	48
EPSON MX-80 SERIAL	50
EPSON MX-80 PARALLEL	51
ANADEX 9500 SERIAL	52
TEXAS INSTRUMENTS SERIAL	54
DRIVE OPTIONS QUME DT-8	56

US MICRO SALES

CHAPTER ONE

ASSEMBLY

1.0 UNPACKING

We have shipped your S100-12 in 2 separate boxes in order to keep within United Parcel shipping requirements. Please keep all boxes and packing materials, should it ever be necessary to return your computer to us.

Please unpack carefully and inspect for any shipping damage. Report any visable damage to your local shippers claims office. Do not attempt to repair any equipment before calling your shipper and having damage inspected. US MICRO SALES is not responsible for shipping damage !.

1.1 INSPECTION

Inspect the S1 MOD assembly for loose screws, connectors, broken wires or wire strands lodged in the bus connector area. Remove if necessary.

1.2 CABINET ASSEMBLY

Remove the cabinet top and set aside. Unpack the front and rear panels from within the bottom half of the cabinet. lay the front and rear panels on the bench in position, be careful as they are wired together. See photo "A". Install 8 captive nuts on each side of the front and rear panels. Using the # 10 phillips screws install the rear panel only, lay the front panel to the right side.

Install the S1 MOD assembly. Use the 6/32 hardware and install the 4 rubber feet. Note the power transformer is positioned to the rear. (Refer to photo B.)

1.3 POWER TRANSFORMER WIRING

Refer to photos "C" & "D" for the correct transformer connections. The wiring shown is for 117 volts. Plug in the reset plug, identified with a red dot, and the reset lamp power, identified with a yellow dot. (Refer to photo E.)

1.4 DISK DRIVE OPTIONS

If you are installing shugart 801 drives received from US MICRO SALES, they have been pre-tested and jumpered as "A" and "B". Each drive is labeled. If you wish to make drive jumper changes, refer to drive notes section 1.12 for jumper options.

1.5 DRIVE MOUNTING

Refer to photo"F", place the drives upside down and mount the hat using the 8/32 hardware and washers provided. Note, this is a preliminary fitting as the final position on the hat will be determined by drive panel tolerance and also how many drives you mount. So do not completely tighten the screws at this time. Final alignment of the drives can be accomplished by repositioning them because the bottom mounting holes have been elongated. Note, because of hat clearance, it is best to install the 1/2 closeout plate if you are only installing one drive. Close out plates are available at no charge from US MICRO SALES. At this time connect the AC AND DC drive power cables. Note the AC drive connections are piggy backed at terminals 1 and 3 of the power transformer. The green ground wire connects to the corner of the S1 MOD under the cabinet ground wire. Note both power connectors are polarized.

1.6 INITIAL POWER

Apply power by connecting AC cord to back of the computer and plugging it into a grounded AC power outlet. Use a meter to check for proper voltages at points shown in photo "E". If voltages are slightly high, but of correct polarity, it may be ok to insert computer boards as the circuit loads will bring them into line. The S1 MOD is designed to run the S-100 + 8 voltage line at + 10 volts. This allows a better power line surge and dropout protection margin.

1.7 BOARD INSTALLATION

Visually inspect all computer boards for any loose parts, or foreign matter on the boards. Remove if necessary. Refer to photo "G" for the correct positioning of the computer boards. We recommend the Disk controller be installed in position 1. Install the CPU and memory boards, be sure to seat firmly. Install the CPU serial cable at the board, note the connector is keyed, install the 50P data cable at the Disk controller and lock in place. Install the 50P cable at the disk drive. Note, the green dot denotes pin 50. refer to photo "G".

Install the RS 232 adapters at the cabinet back, if you are making any special jumper changes on the printer card, it will be easier to make these changes before installation to the cabinet. Use the 4/40 hardware provided. (Note "B" will be the CP/M serial terminal port and "A" is the CP/M serial lister printer port.)

1.8 SERIAL TERMINAL

At this time connect a serial terminal to port "B". Upon power on a sign on message should appear on the terminal. If no sign on message appears, power down and refer to step 1.9. Manually reset the computer to verify the sign on message comes on each time. At this time you may operate the computer in monitor mode. Refer to the general operating manual for the monitor commands.

1.9 COMPUTER TERMINAL CHECKS

Verify the connections of all cables, and the correct positioning of all boards. Check your RS 232 serial terminal cable. Pins 1 through 8 and 20 are connected in the standard cable. Following instructions provided with your terminal, check it in local or self test mode. Verify that the terminal baud rate is set for 9600 baud. Make sure the terminal serial cable is connected to the correct port on the terminal, as some terminals have auxilliary or printer ports. If the terminal checks ok in self test and the cable is ok, reinstall the boards, and cables.

1.10 DISK OPERATION CHECK

If disk drives are installed, verify that the motors are running. Insert the system diskette with the label to the left, and close the drive door. Perform a "control C". Note systems with hard disk installed will boot with a "control F". They should boot into CP/M with an A>. If you do not hear the head load or see the drive access light come on, remove the diskette and check the following:

1. Manually reset the computer
2. Proper system diskette installed correctly?
3. Diskette in drive "A" ?
4. Are the motors turning?
5. Is the drive ready light lit?
6. Is the 50P cable correctly installed?
7. Are both AC and DC cables correctly seated?
8. You may wish to re-configure the drives as "B" to "A" to determine if one drive is at fault.

If you achieve a proper system boot up, refer to your systems operation manual for proper back up procedures as well as verification of drive "B" operation.

For the AC connections for 220 Volts, Refer to photo "C".

1.11 FINAL CLOSEOUT

If you are satisfied with proper operation of all functions, you may install the top cabinet.

1.12 DRIVE OPTIONS SHUGART 801

T-1 thru T-6 are jumpered for termination. T-2 is a spare terminator for radial head load. T-2 is jumpered on both drives A and B. T-1 is the terminator for drive select and should be jumpered on the drive at the end of the 50 pin cable, (drive B). T-3, T-4, T-5 and T-6 are termination jumpers for multiplex inputs and are jumpered on the drive at the end of the 50 pin cable (drive B). If only one drive is hooked up, T-1 thru T-6 should be jumpered. On multiplex inputs only T-2 should be jumpered. On multiplex inputs only T-2 should be jumpered and the last drive on the cable (drive B) should have T-1 thru T-6 jumpered for termination.

DS1 thru DS4 are for "drive Select". DS1 is jumpered for drive A and DS2 is jumpered for drive B.

HL and DS jumpers are for "stepper power from head load" and "stepper power from drive select" respectively. All drives are sent jumpered DS.

Y and Z are jumpers for "in use from head load" and "in use from drive select" respectively. All drives are sent jumpered Y.

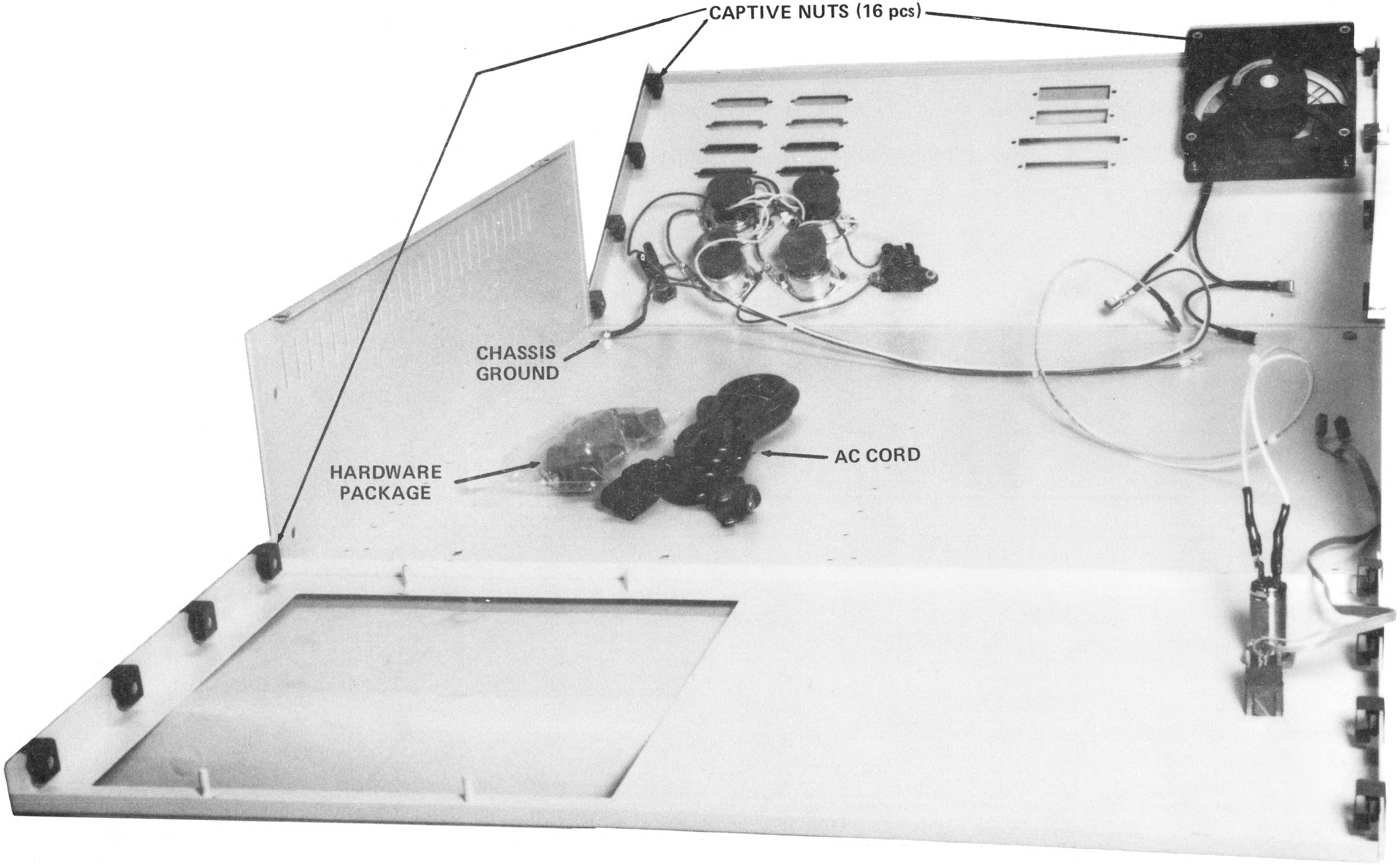
D and DC are jumpers for "alternate input-in use" and "alternate output disk change" respectively. These are sent open with all drives.

A, B, X, are jumpers for "radial head load" A and B are sent jumpered on all drives. X is sent open on all drives.

C is a jumper for "alternate input-head load" All drives are sent jumpered.

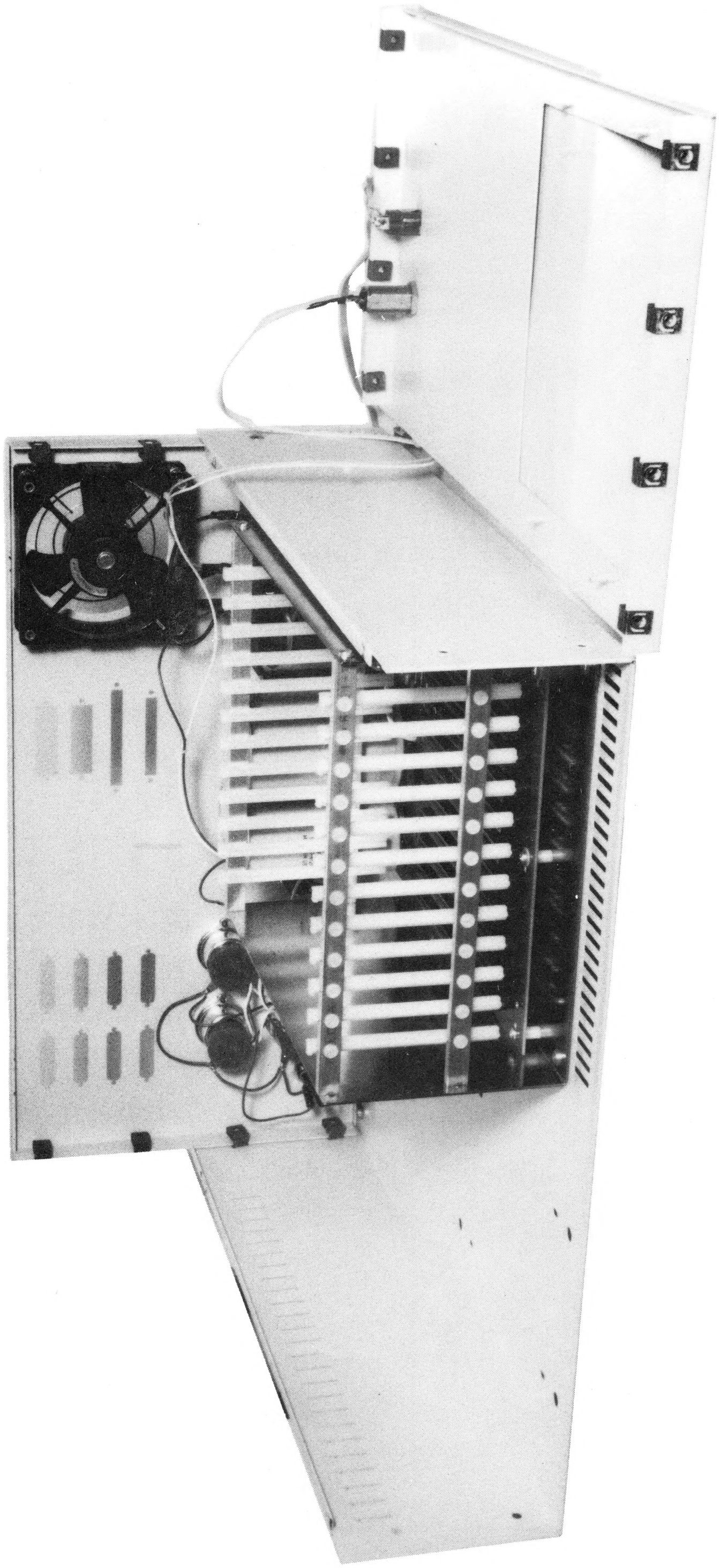
800 and 801 are jumpers for "soft sector" and "hard sector". all drives are sent jumpered soft sector.(800)

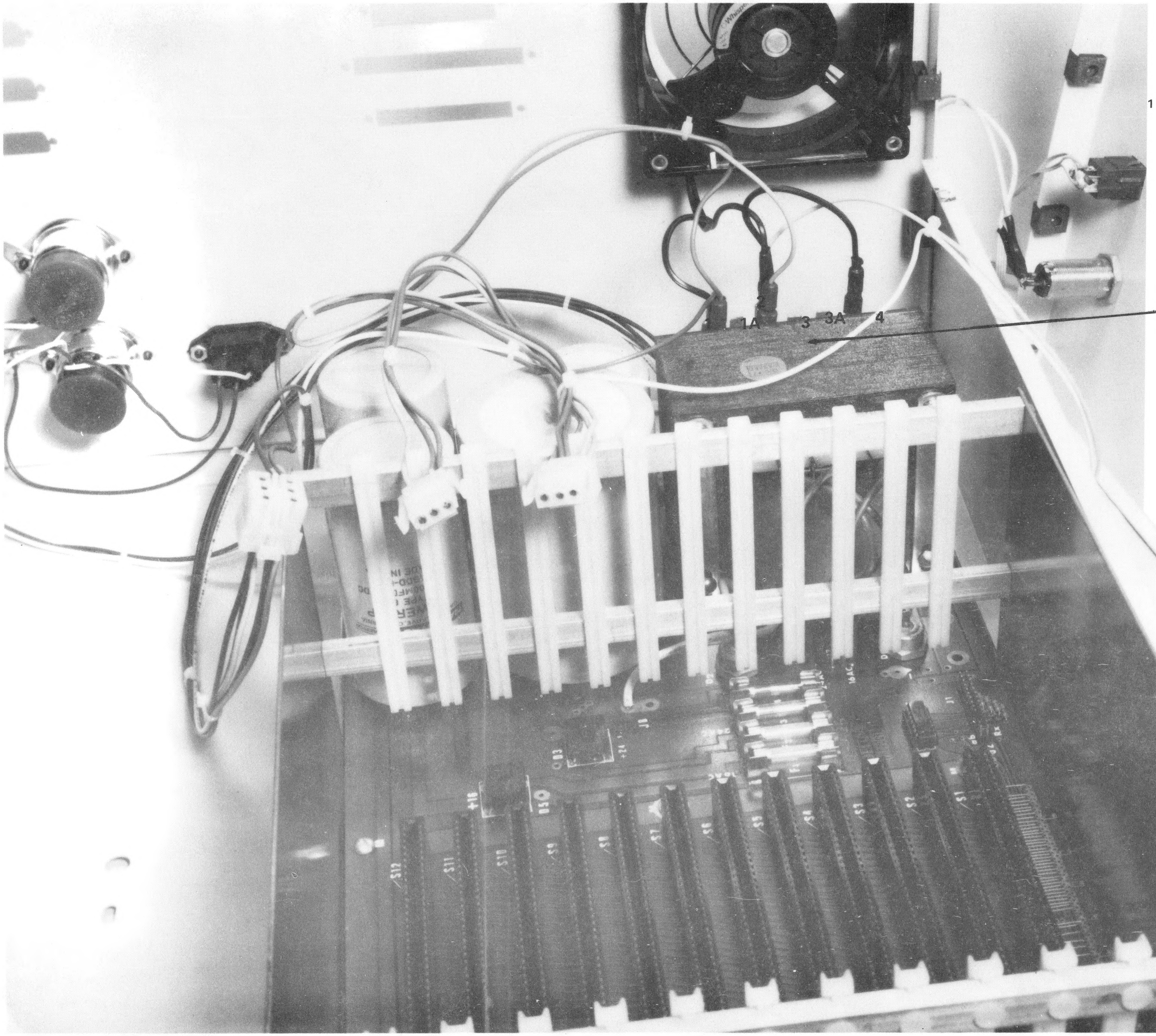
NOTE: SEE APPENDIX FOR DRIVE NOTES ON THE QUME DT-8.

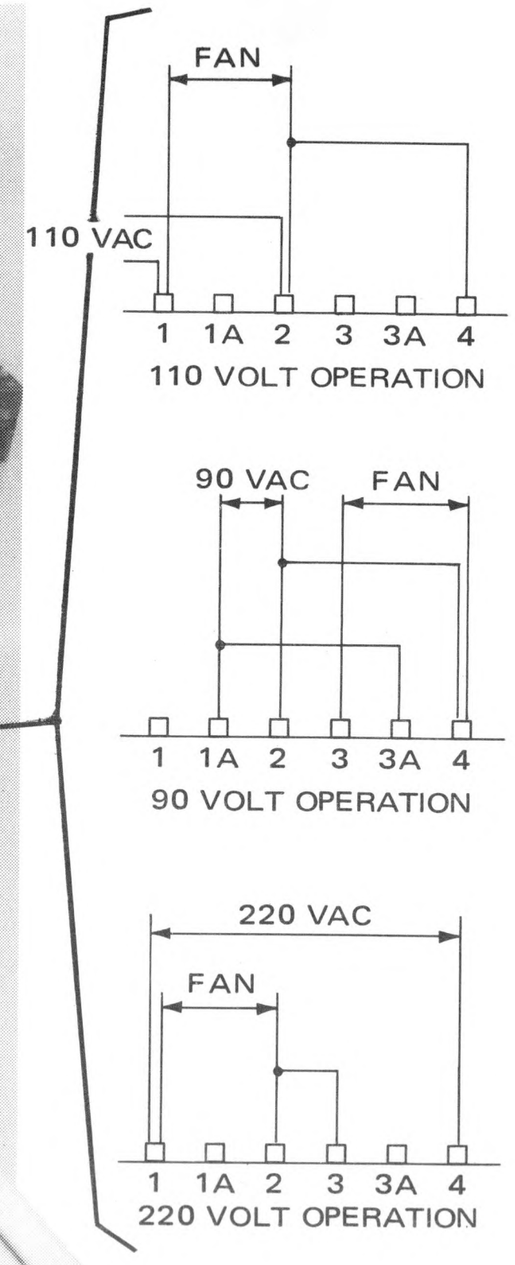

CAPTIVE NUTS (16 pcs)
CHASSIS GROUND
HARDWARE PACKAGE
AC CORD

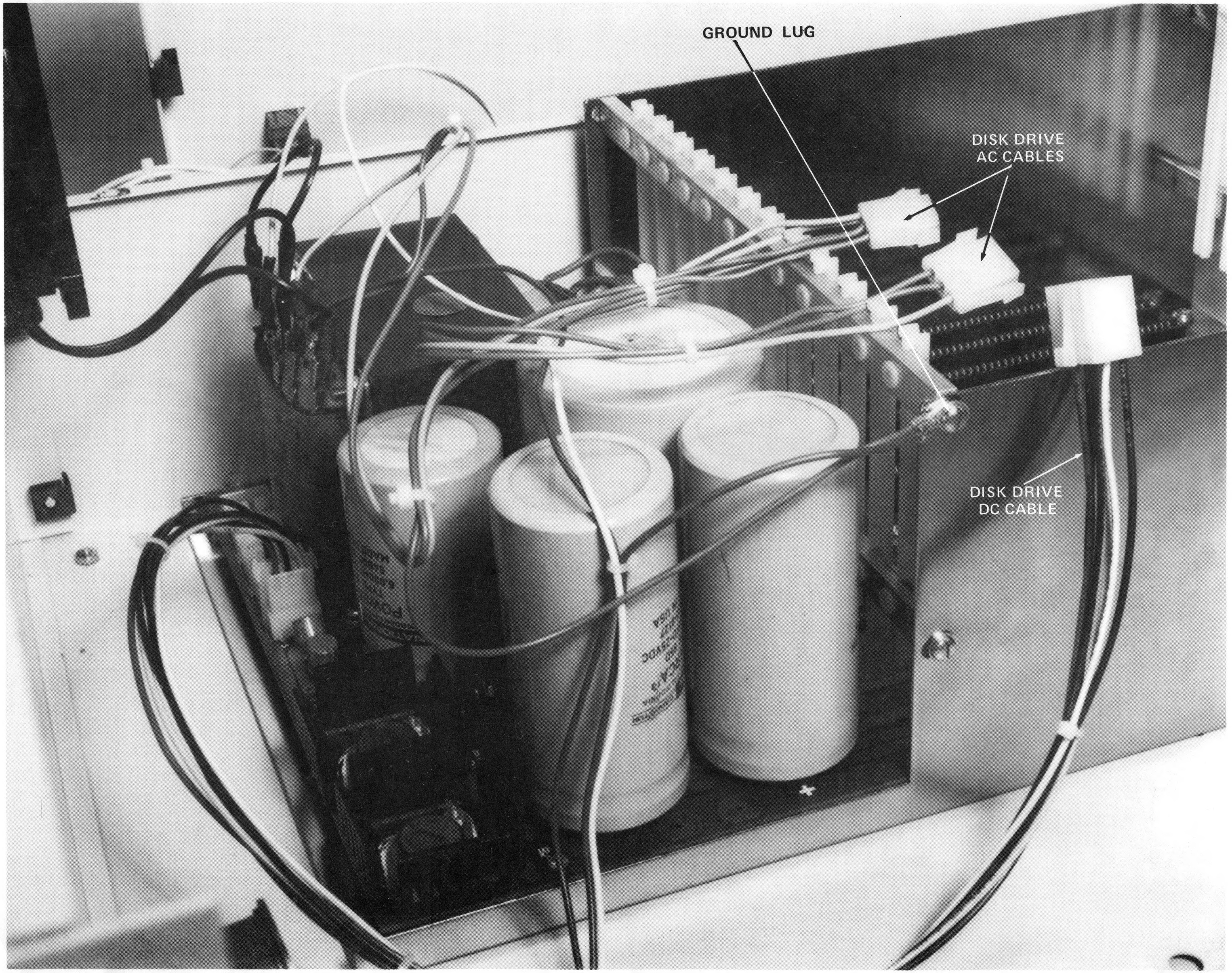
"A"

"B"

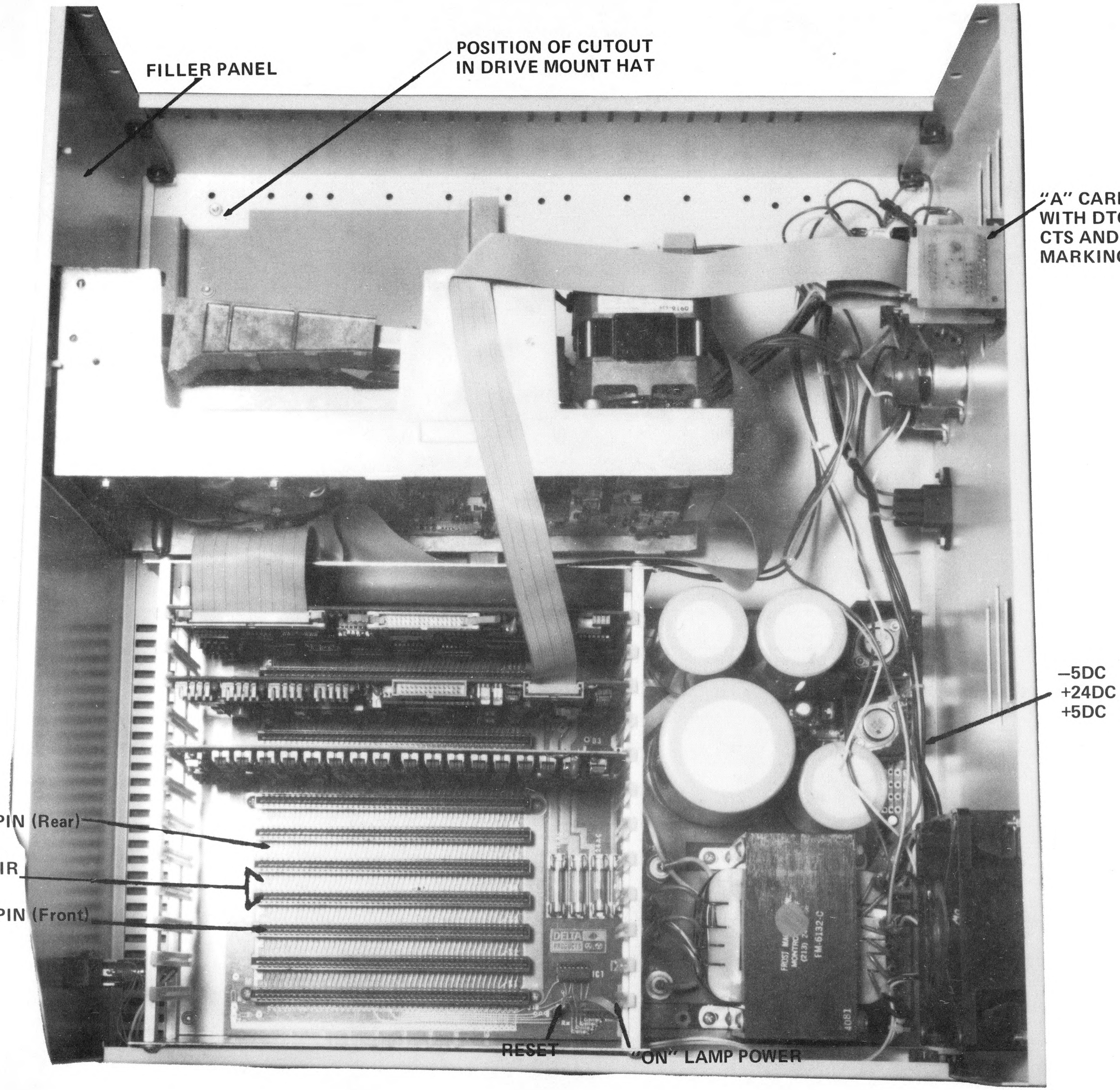


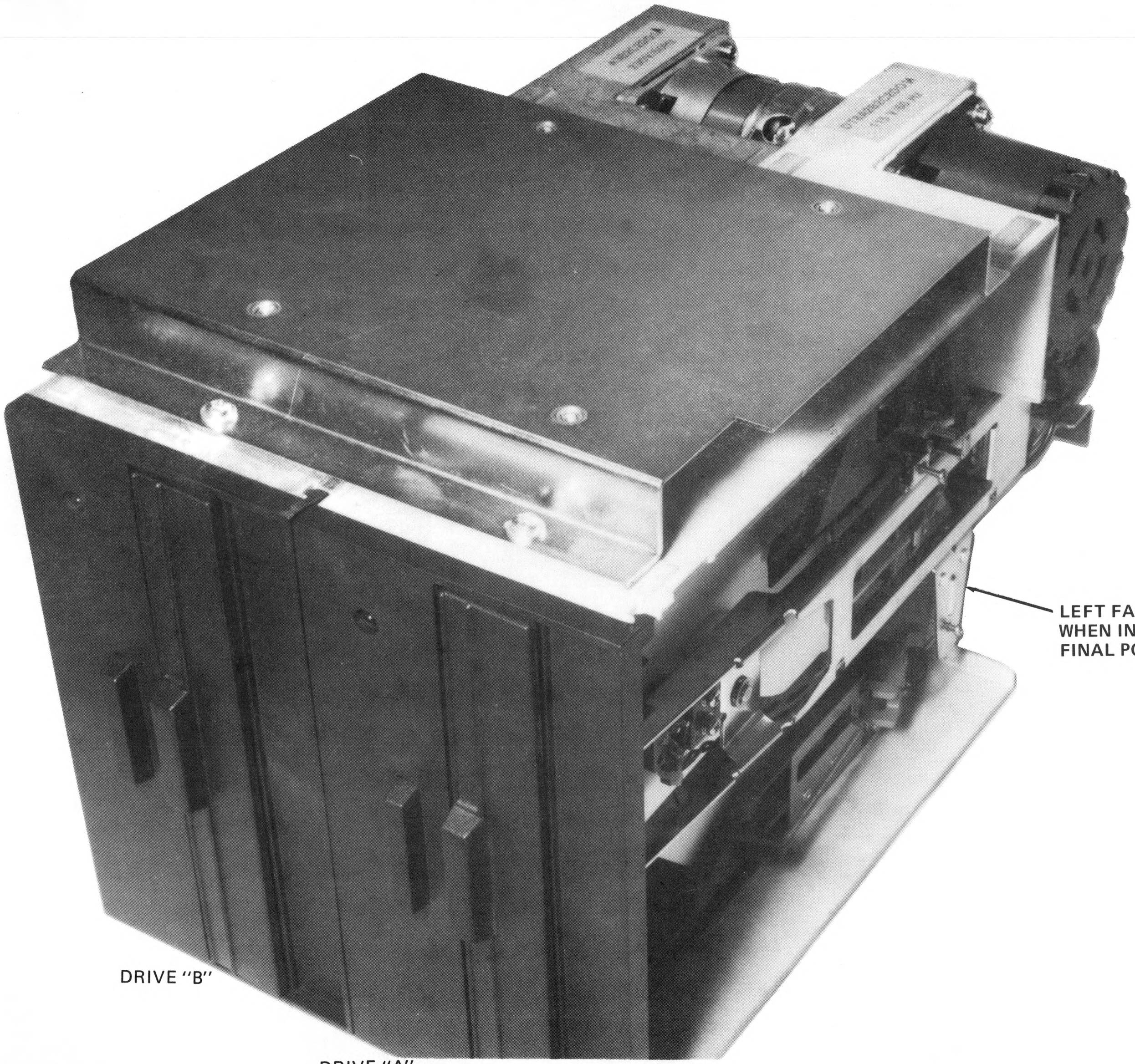

1A
3
3A
4


FAN
110 VAC
1 1A 2 3 3A 4
110 VOLT OPERATION
90 VAC
FAN
1 1A 2 3 3A 4
90 VOLT OPERATION
220 VAC
FAN
1 1A 2 3 3A 4
220 VOLT OPERATION



"D"


FILLER PANEL
POSITION OF CUTOUT
IN DRIVE MOUNT HAT
"A" CARD
WITH DTC, RTS
CTS AND DSR
MARKINGS TOPSIDE
–5DC
+24DC
+5DC
SECOND PIN (Rear)
–18VDC
FIRST PAIR
+10VDC
SECOND PIN (Front)
+18VDC
RESET
"ON" LAMP POWER



QUME DT-8
SHOWN

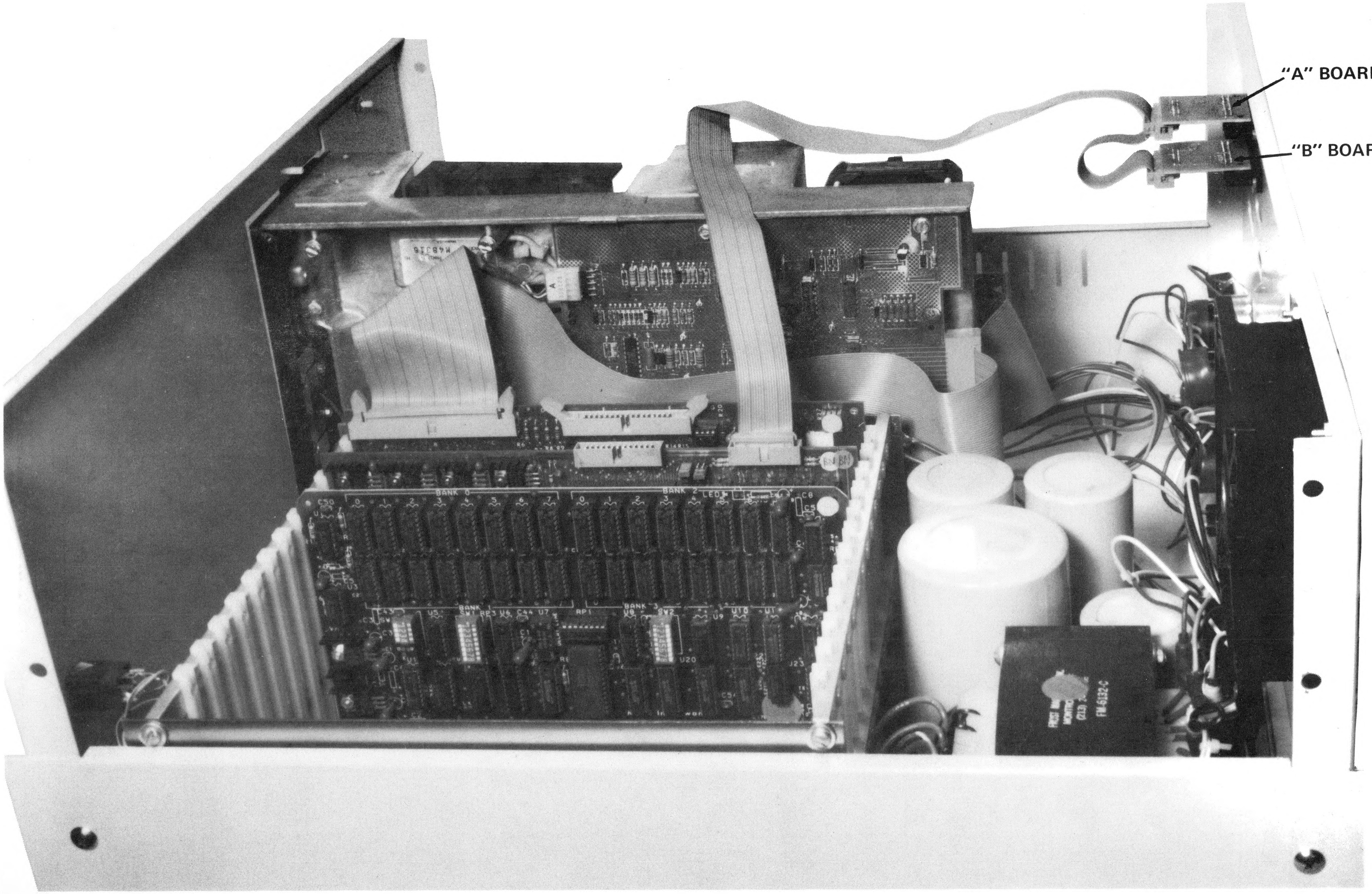
LEFT FACE
WHEN IN
FINAL POSITION

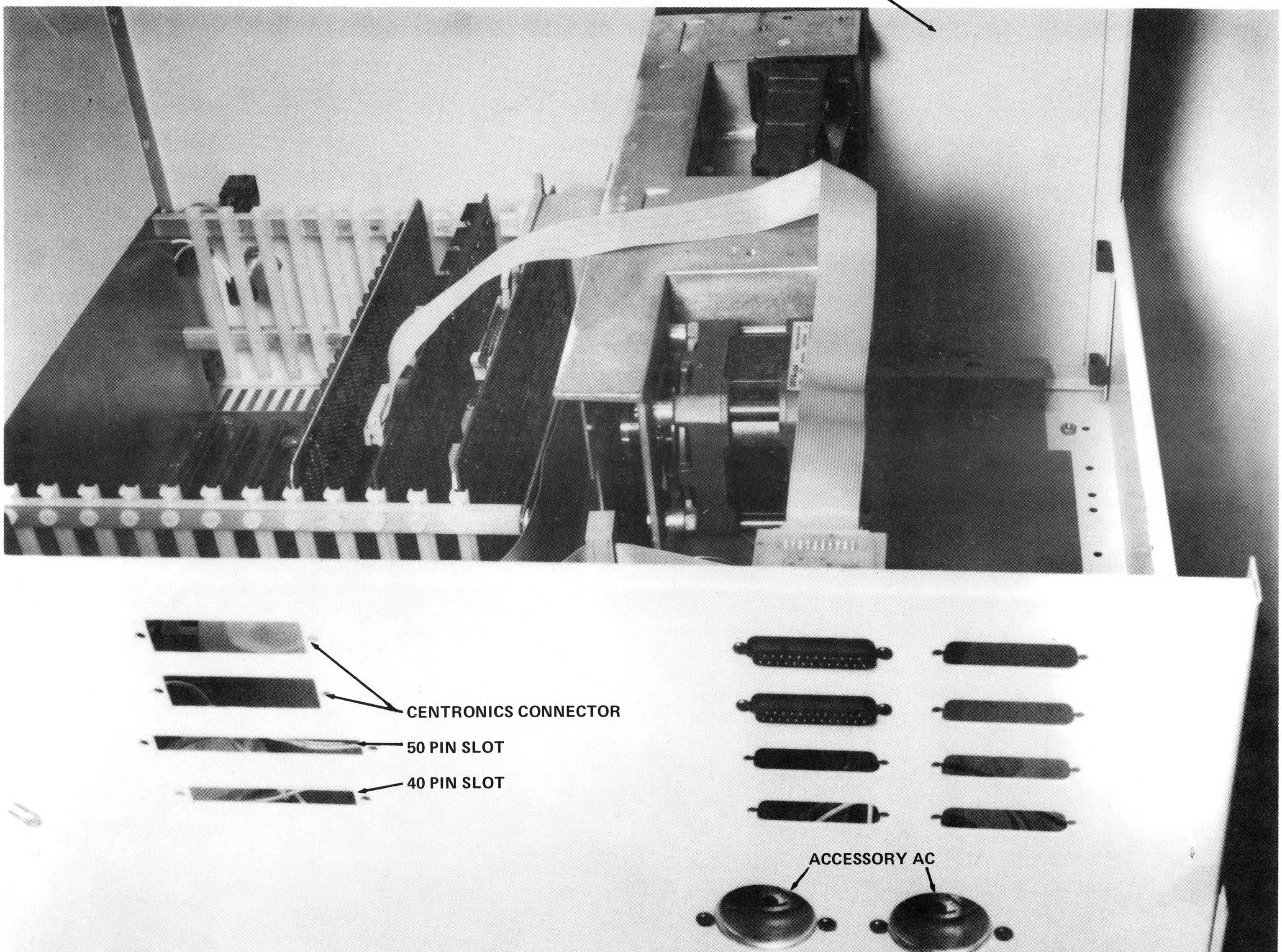
DRIVE "B"

DRIVE "A"

"F"

TYPICAL BOARD PLACEMENT




½ SLOT PANEL FILLER IN PLACE
CENTRONICS CONNECTOR
50 PIN SLOT
40 PIN SLOT
ACCESSORY AC

„H„

CHAPTER TWO

GENERAL OPERATING PROCEDURES

2.1 SHOCK AND TILT SENSORS

Your computer system may have been sent to you with shock sensors attached to the outside of the shipping carton(s). This sensor is a rectangular orange and white piece of paper with a small glass tube in the center. The glass tube is designed to display a red liquid inside when the shipping carton has been subjected to an impact which approaches the shock absorption capabilities of US MICRO SALES packaging materials.

If any package that you receive has shock sensors which are red, write the following on the bill of lading: "Shock sensor broken - possible shipping damage". This statement should be initialed and dated.

Some XOR computer systems can become damaged if excessive force is applied to them. If you should find that the system does not function properly and the sensor has been broken, it is the responsibility of the shipper and you to come to agreement on resolving payment for the damages. A claim must be filed with the shipping firm in your local area BEFORE the equipment can be shipped back to US MICRO SALES (See RMA Procedure in a later section entitled "Service Policy")

The "Tip-N-Tell" sensor is a rectangular red tag with an arrow-shaped cutout in it. The arrow is positioned on the shipping carton to act as a "This End Up" indicator. There are some blue crystals which fill the stem end of the arrow. Should the shipping carton be tilted during shipment to an angle greater than approximately sixty degrees, the blue crystals will spill out and remain out, turning the arrow-shaped cutout blue.

Again, if the "Tip-N-Tell" sensor has been tripped make sure to note the fact that the package has been inverted and that possible shipping damage exists on the bill of lading.

If any damage is apparent from the outside of the box or any damage is found to exist inside of the box after opening, be sure to save all packing material. DO NOT TRY TO FIX BROKEN UNITS WITHOUT FIRST NOTIFING THE FACTORY. You may limit your damage claims by modifing or trying to fix equipment that has been broken.

2.2 HARD DISK SYSTEMS

All of the Hard Disk Systems now produced by US MICRO have some form of locking device holding the heads and/or the spindles in place during shipment. Make sure to find the manual on the system that you are using (it will be included in the documentation package shipped with the system) and READ IT THOROUGHLY BEFORE ATTEMPTING TO TURN ON THE POWER. Turning on the power without releasing the head or spindle locks could result in irrepairable damage to the unit.

2.3 POWER UP

The computer system has a resident monitor PROM on the CPU board. The monitor PROM initializes the 8251 serial controller chips in the following way: 9600 baud, 8 data bits, 1 start bit, 1 stop bit, and no parity. The computer is set up to look like a MODEM (i.e. pin 2 on the RS232 connector is transmit data, pin 3 is receive data, and pin 7 is signal ground).

Turn the key to the "ON" position, turn the green POWER switch ON and momentarily push the RESET button down. Wait for approximately half a second. The computer is running self-diagnostics. When it has finished checking itself and its system memory, it will sign on with a message saying, "XOR SYSTEM MONITOR..." and it will give a number in hex (usually this number is F800).

If you are running a computer with 64K of RAM memory, and the number on the screen is anything different than F800, shut off the computer and contact an authorized XOR service person.

The monitor PROM resides in the very top 2K of system memory. It is not permanently there but is switched out when the disk operating system is loaded in. It is the design of the CPU that memory underneath the PROM can be written to even while the PROM is in place. Further information about the functions of the PROM may be found in the CPU manual or the manual entitled "Special Hardware and Software Conventions".

It is very important for the computer to come up in an intermediate state before attempting to bring in the CP/M operating system. In this way, if there is a problem with the computer you can see it before you have damaged valuable information on the floppy disk.

Certain utility programs are available to you from the PROM to correct a problem by allowing the computer fix itself. The diagnostics that are run are generally enough to assure you that the computer is going to at least be able to load the operating system from the floppy disk without "crashing." Once the operating system has loaded, the more sophisticated debugging tools available on the disk can be used to locate and/or correct any further problems.

2.4 MONITOR COMMANDS

There are several monitor commands that are available for looking into the computer or for modifying and viewing memory locations.

DDUMP - The dump command will accept 4 hex bytes, jump to the next field, accept 4 more hex bytes and then display all the of the memory locations between those two addresses. If you should wish to enter the numbers without leading zeros, you may do so by hitting a carriage return. For example, you could dump location 2 to location 8 in the computer by typing D2 <CR> followed by 8 <CR>.

L (LOAD) - The load command will accept one 4 byte address and then display the contents of that memory location on the screen. The prompt character will allow you to replace what is in that location of memory with the data that you type onto the screen. If you merely want to look at system memory, you can enter a carriage return or a series of carriage returns. The memory locations will not be modified, but only displayed on the screen in sequencial order.

F (FILL) - The "FILL" command accepts a 4 byte (starting) address, moves to the next field, accepts a second 4 byte (ending) address, moves to the next field and accepts a 2 byte set of data. At this point, the "FILL" command will automatically be executed, filling all system memory between the starting address and the ending address with the data character set.

M (MOVE) - The "MOVE" command operates in the same mannner as the "FILL" command with the exception that the 4 byte addresses are, respectively, the address of the source data and the address of the memory location to which the source data is to be moved. The 2 bytes which are entered next indicate the size (number of bytes in hex) of the block of memory to be moved.

V (VIEW) - The "VIEW" command upon receiving a 4 byte starting memory location will display sixteen lines of ASCII data (640 bytes) on the CRT screen. Typing carriage returns following the initial display causes the next sequential 640 byte blocks to appear.

2.5 DISK BOOT COMMANDS

There are three disk boot commands currently being utilized in systems manufactured by US MICRO SALES They are: control C (^C), control F (^F), and control A (^A).

NOTE: TO EXECUTE ANY CONTROL FUNCTION, THE CONTROL KEY MUST BE HELD DOWN WHILE YOU TYPE THE APPROPRIATE CHARACTER KEY.

Once the monitor has signed on and the diagnostics have presented an "F800" to the CRT (the last memory location tested). Insert a floppy disk into the floppy disk drive "A" and close the drive's door. If your computer system contains a XOR floppy disk controller board, typing ^C will load the CP/M operating system off of the disk into the computer and allow you to begin operation under CP/M. If your XOR computer system contains hard disk drives as well as floppies, typing ^A will boot the operating system on the first tracks of the the hard disk drive, typing a ^F will boot the operating system from the floppy.

More detailed explanations of the procedures for operating two disk controllers in the same computer system are contained in the maintainence section of the "Special Hardware and Software Conventions" manual.

2.6 BOOTING THE OPERATING SYSTEM

If you have never operated a XOR computer system before, it is best to try to bring it up first on a floppy disk. The floppy disk portion of the system is the most reliable and likely to function if there has been some unapparent shipping damage to the system. Also if, for some reason, you or the system destroy the information on the floppy disk, it is easily replaced. Replacing the information on a hard disk usually requires more time and takes more expertise to accomplish.

One of the first tests you should perform while getting familiar with your computer is to momentarily push the reset switch. Then, without inserting a diskette into the floppy disk drive, hold down the "CONTROL" key on the terminal keyboard and type a "C". If the system is functioning normally, an "80" error code will appear on the CRT screen.

With your computer system you have received two or three diskettes. These diskettes are your master operating system diskettes. They contain your utility programs and the programs which are the parts and pieces needed to generate and/or modify the operating system of your computer. If your computer was designed to run CP/M only, you will have received two diskettes (Disk A & B). With MP/M computer systems, you will have received a third diskette (Disk M).

The "A" diskette contains approximately twenty five programs. The programs on this disk are the tools and utilities of CP/M (e.g. PIP, DDT, ED, ASM, SUBMIT). The programs on the "B" disk are the subroutines, assembly files, and the equate tables with which the operating system is generated and/or modified. The "M" disk contains subroutines and utility programs specially designed to run under MP/M. A listing and brief description of the contents of these diskettes can be found in another section of this manual (See Table of Contents). The use of these disks and their programs is explained in this manual and the Digital Research manuals included in your documentation.

Insert the "A" diskette into the floppy disk drive (if two drives are available use the one on the left as you face the system). The label on the diskette should be on the left face of the diskette along the edge farthest from the face of the drive and the label's long edge should be parallel to the face of the drive. Insert the disk firmly and allow it to stay in place before closing the door. If the door is difficult to close or you hear the sound of crunching plastic or paper, do not force the door closed. Take the disk out and reinsert it making sure to push it all the way to the rear of the slot in the drive before closing the door. Having accomplished this, hold the "CONTROL" key on the terminal keyboard down and type C (or F,etc.).

If the computer does not boot within approximately half a second and display a "sign on" message, reset the computer by momentarily pushing the reset switch down, waiting for the "sign on" message to appear on the CRT and once again typing the ^C.

Perhaps the most attractive feature of having a system monitor appear before the system attempts to boot the floppy disk is that the system is able to do reporting of errors should the disk controller not be able to come up. An "80" error code means that the disk drive is not ready. Other typical errors (e.g. "10", "20" and "08") can be deciphered by reading your orange XOR Disk Controller Manual.

If the CP/M operating system has signed on successfully, you will now have on the CRT a statement indicating the operating system version number and last date of modification, the system size and the type of diskette off of which the operating system was read. At this point, if you are unfamiliar with CP/M commands, you should start reading the manuals from Digital Research (beginning with the "CP/M Features and Facilities Guide") which are included in the documentation package received with your system.

Digital Research's manuals are not written for beginners. D.R. makes the tacit assumption that the person reading the manual has already had some experience in the operation of microcomputers. There are several books that are much, much better for learning how to use CP/M. The best of these is "The CP/M Handbook" by Rodney Zacks. XOR stocks this book and it can be ordered through the Sales Department.

It is our experience that quite often while a person is learning to use a computer system, he will occasionally make a mistake which destroys a master operating system diskette received with the computer. We, therefore, suggest that you format a new floppy disk and copy the master disks onto it before proceeding any further (See your "XOR-DSK Operation Manual" for the correct operation of the DFOCO, Double density FOrmat and COpy, program). It is wise to make four or five copies of the operating system disks for future use and to number the copies so that you will be able to identify a damaged disk and refrain from using it.

2.7 BACKING UP

Computers are an extremely powerful tool, but there are times when problems occur. The inconvenience that occurs when a computer "goes down" after you are relying quite heavily on it can be quite costly. There is one sure way to minimize that cost - backing up on a regular basis. Backing up means copying to portable, external media (usually floppy diskettes) the vital information stored in the computer's memory system.

This may seem like a redundant procedure since the data you are copying is already stored in the computer's memory. However, the time to back up the data is much, much less expensive than the time it would take to re-enter that data should your computer system "go down". If you have two to four hours worth of work in the computer, spending two to three minutes to back it up is a small insurance premium to pay.

It is best to have to have more than one set of back up diskettes. The computer could "go down" while it is backing up data. In that case, the original disk and the backup disk could both be damaged. A wiser method of doing a backup is having a father/son type of backup system where you back up one time using disk set "A", and back up the next time using disk set "B". Therefore, if something should go wrong while you're backing up, you would have data available from the previous back up.

The preferred method of backing up in the CP/M environment is bundling all of the pertinent files and backing them up all at one time using the "SUBMIT" utility of CP/M. In this way, each department of a company could back up its own data on a single diskette (or a single cartridge tape if you're using the XOR -17 tape drive). The accounting department would have its own set of back up disks as would the inventory control department, the sales department, etc.

2.8 SAFETY PRECAUTIONS

1. Before operating the equipment, remove the top from the computer and look for any wires, boards, and/or components that may have become dislodged during shipment. Take the fuse out and make sure that it matches the voltage and the amperage rating for the system that you are using.

2. Do not overload the external power outlets located on the back of the mainframe. They are designed to handle no more than 100 Watts per connector, a total of 400 Watts.

3. Do not leave the computer running unattended for any great length of time.

5. Do not operate the computer with the cover removed. There are two reasons why this should not be done. Exposed high current/high voltage electrical sources could present a safety hazard if someone were to stick their hand inside the computer. The other reason is that the cooling system of the computer is designed to create certain air flow patterns that can only be maintained as long as the top is on the system. Operating the computer with the top removed can result in the overheating of critical components in the system.

6. Do not modify any internal electrical wiring. The wiring that is inside the computer has been specifically designed for the voltage and amperage requirements of this system. Cutting wires or adding components to the computer may exceed the ratings of the system. All warranties are void if any portion of your computer system is altered by anyone other than factory authorized personnel.

2.9 SERVICE POLICY

1. Service will be performed on XOR equipment at The plant facilities in Huntington Beach, or at an authorized XOR Service Center.

2. Equipment will be accepted for service only after you have been assigned a "Returned Material Authorization Number". (RMA#). This number may be obtained by calling the US MICRO Sales, Sales Department. The Sales Department will need to know the following information before an RMA# can be issued.

A) - The number of the invoice on which you were billed for the equipment that you wish repaired.

B) - The serial number of the system, if one has been assigned.

C) - A detailed description of the problem with the equipment and the results of any diagnostic tests which you may have performed. If a written description of the problem is not received with the returned merchandise you will be subject to a labor charge of $50/hr for the time it takes our technicians to diagnose the problem.

3. All shipments without our RMA number clearly marked on the outside of the shipping container will be returned to the sender.

4. Warranty (6 Months parts/90 days labor, from the date of purchase) repairs will be made at no charge for parts and/or labor for XOR hardware only. Merchandise such as drives, terminals, and printers sold by US MICRO SALES as part of their systems will be subject to the manufacturer's warranty and will usually be sent back to the manufacturer for repair. Repair and freight charges for non-XOR merchandise will be billed to you by US MICRO SALES. All warranties are void if any portion of your computer system is altered by anyone other than factory authorized personnel.

5. Incoming freight MUST BE PREPAID. US MICRO SALES will not accept incoming freight which is marked COLLECT or COD. Shipping charges for the return of repaired equipment to you that is under warranty will be prepaid by US MICRO SALES except in cases where you specify method of shipment other than the one chosen by US MICRO SALES. All return shipping charges for equipment that is no longer under warranty are the responsibility of the party who sent the equipment for repairs.

2.10 ** DISK A - XOR DISK UTILITIES **

DFOCO:

DFOCO is a fast format and copy routine which allows the user to copy and format in both single and double density and allows different sector sizes to be utilized. (128 , 256 , 512 , 1024 bytes per sector)

(More information concerning DFOCO can be found in the "XOR-DSK MANUAL" and in the "SPECIAL HARDWARE AND SOFTWARE CONVENTIONS MANUAL" under section "SU". On the distribution disk the file named DFOCO.DOC will contain more information.)

M2:

M2 is a non destructive memory test which can be left to run for long periods of time on its own. M2 continuously reexecutes itself from the beginning once a pass is completed.

(More information concerning M2 can be found in the "SPECIAL HARDWARE AND SOFTWARE CONVENTIONS" MANUAL UNDER "SD")

M:

M is a non destructive memory test which can be left to run for long periods of time on its own. M will not restart from the beginning once it completes a pass, but will continuously run random number test. For more information on M refer to the M.DOC file on your disk.

(More information concerning M can be found in the XOR "64K" manual and in the "SPECIAL HARDWARE AND SOFTWARE CONVENTIONS" manual under "SD")

M48:

M48 is a non-destructive memory test for systems with mulitiple memory banks. This test will test one bank at a time. It is simular to M and M2.

WORM:

WORM is a destructive memory test for all of RAM which will run continuously. The only exit from this routine is to RESET the unit.

(More information concerning WORM can be found in the XOR "64K" manual, and in the "SPECIAL HARDWARE AND SOFTWARE CONVENTIONS" manual under "SD")

WORM48:

WORM48 is the combination of both M48 and WORM all in one program. It executes the WORM program in alternate memory banks sequentially starting with bank 0. The program continously displays the address and bank the program is operating in so if it were to find a bad spot in RAM it would stop with the last displayed address on the screen. This test can be left running for long periods of time.

DUMP & DDUMP:

DUMP & DDUMP is a disk read utility. The user can write to the disk as well as using the EDIT mode.

(More information concerning DUMP & DDUMP can be found in the "XOR-DSK" manual)

DSYSGEN:

DSYSGEN is used to copy a developed system onto a disk (hard disks as well as floppys).

(More information concerning DSYSGEN can be found in the XOR "SPECIAL HARDWARE AND SOFTWARE CONVENTIONS" manual and the section entitled DSYSEGEN STEP-BY-STEP in this manual.)

D:

D is a directory utility which will print to the console all files on disk in alphabetical order and their file size. At the end of the directory the total number of files and their total size is also printed to the console. Other drives can be referenced example "D B:"<CR> "D"<CR> "D B:*.ASM"<CR>

DSKTYP:

DSKTYP is a disk identification program. The program asks the user which drive to check, once answered the program will print to the console what type of diskette is in the specified drive.

DDT:

DDT is a utility which has many features one of which allows the user to read .COM and .HEX files into RAM for execution. Under DDT the user may edit, examine, or link .COM or .HEX files or make small routines for test or other purposes. If under DDT you wish to save an edited program, use the CP/M SAVE NN FILENAME.TYPE command. (NN being the number of records to be saved)

(More information concerning DDT can be found in the DIGITAL RESEARCH'S "CP/M DYNAMIC DEBUGGING TOOL (DDT)" manual.)

ED:

ED is a source editor for all source files.

(More information concerning ED can be found in the DIGITAL RESEARCH'S "ED: A CONTEXT EDITOR FOR CP/M DISK SYSTEM" manual.)

ASM:

ASM is a 8080 source code assembler. This utility makes a .SYM .PRN .HEX file from .ASM files.

(More information concerning ASM can be found in the DIGITAL RESEARCH'S "CP/M ASSEMBLER (ASM)" manual.)

LOAD:

LOAD takes the .HEX files developed from source file and makes a .COM for execution of the file.

STAT:

STAT is a utility which will print to the console the size of any file or the amount of storage remaining on the disk. Example "STAT"<CR> "STAT B:"<CR> "STAT *.*"<CR> "STAT B: STAT.COM"<CR>

(More information concerning STAT can be found in the DIGITAL RESEARCH'S "CP/M 2.0 USER'S GUIDE FOR CP/M 1.4 OWNERS" manual.)

PIP:

PIP is a copy utility which will copy disk to disk from one to all files on the source disk. Example "PIP B:=A:PIP.COM"<CR> "PIP B:=A:*.*"<CR>

(More information concerning PIP can be found in the DIGITAL RESEARCH'S "CP/M 2.0 USER'S GUIDE FOR CP/M 1.4 OWNERS" manual.)

SUBMIT:

SUBMIT is a special utility which allows console input simulation so that long tedious steps which have to be repeated quite often can be done by typing a one line command. Example submit file.

```
DDT
IEXAMPLE.HEX
R
G0
SAVE 1 EXAMPLE.COM
```

XSUB:

XSUB extends the power of SUBMIT to include line input to programs as well as the console commands, the XSUB when included must be in the first line of the sub file.

(More information concerning XSUB can be found in the DIGITAL RESEARCH'S "CP/M 2.0 USER'S GUIDE FOR CP/M 1.4 OWNERS" manual.)

MACRO.LIB & Z80.LIB

MACRO.LIB & Z80.LIB are the library files for DIGITAL RESEARCH'S MACRO.COM.

TIP:

TIP is the utility used with the XOR TAPE DRIVES.

(More information concerning TIP can be found in the "XOR-17 MEG OPERATION MANUAL" manual.)

PFMT:

PFMT is the format program for the PRIAM HARD DISK.

SAFMT:

SAFMT is the format program for the SHUGART HARD DISK.

QFMT:

QFMT is the format program for the SMS 10, 20, 30, 40 MEG HARD DISK.

NOTE: PFMT, SAFMT, QFMT are DANGEROUS programs. They will completely erase all data including the operating system on the disks. There is no way to recover that data once the format has been completed.

2.11 ** XOR DISK B "SYSTEM SOURCE DISK" **

STDSYS60.SUB:

STDSYS60 is the file used to develop standard 8" floppy systems.

MINSYS60.SUB:

MINSYS60 is the file used to develop the mini 5" disk systems for both the 96 TPI & 48 TPI drives. The file named MINEQU60 should be checked to suit your system needs.

BLKSYS60.SUB:

BLKSYS60 is the file used to develop 8" & 5" disk systems with BLOCKING/DEBLOCKING MULTI SECTORS.

PRMSYS60.SUB:

PRMSYS60.SUB is the file used to develop the PRIAM 3350 HARD DISK system.

DTCSYS60.SUB:

DTCSYS60 is the file used to develop SHUGART 10 MEG drive systems.

SMSSYS60.SUB:

SMSSYS60 is the file used to develop SMS 10, 20, 30, 40 MEG drive systems.

DMPM2SYS.SUB:

DMPM2SYS is the file used to develop MP/M systems for the assorted systems mentioned above.

DSYSGEN.COM:

DSYSGEN is the file used to put the predeveloped systems onto the system track of the disk.

STDEQU60.ASM:

STDEQU60 is the equate table used in the development of the STANDARD 8" SYSTEMS. Normally in this file just the "MAXI8" is set to TRUE (DPROM should always be set true for any system). The SEEK RATE for the drives is determined by the type of drive used in the system, a SEEK RATE of "0" is normally set for QUME DT-8 or SHUGART 850, a setting of "1" is normally set for the SHUGART 801.

MINEQU60.ASM:

MINEQU60 is the equate table used in the development of all MINI 5" SYSTEMS. The proper setting for the system is dependent on the type of drives, a 96 TPI mini drive will have the "MINI96" set TRUE, a 48 TPI mini drive will have the "MINI48" set TRUE. The "MAXI8" can be set FALSE if you desire not to include 8" drives in your system.

BLKEQU60.ASM:

BLKEQU60 is the equate table used in the development of any system requiring BLOCKING/DEBLOCKING MULTI SECTOR operation. In doing this "DBLOCK" must be set TRUE, along with the desired sector size, so either "F512" and or "F256" must be set TRUE. The desired setting for the floppys must also be ajusted. Both "MAXI8" and/or "MINIXX" can be set TRUE. The "SKEW" can also be set TRUE to gain speed from the floppys. (this setting can also be set TRUE in any other equate table)

DTCEQU60.ASM:

DTCEQU60 is the equate table used in the development of the SHUGART 10 MEG systems. The "DBLOCK" is normally set TRUE, as is the "MAXI8".

SMSEQU60.ASM:

SMSEQU60 is the equate table used in the development of the SMS 10, 20, 30, 40 MEG systems. The settings for the different drives are as follows:

1)An SMS 10 MEG system;

```
QUANTUM EQU  TRUE   ;This always set true for any SMS
SMS10   EQU  TRUE
SMS20   EQU  FALSE
SMS30   EQU  FALSE
SMS40   EQU  FALSE
```

2)An SMS 20 MEG system;

```
QUANTUM EQU  TRUE
SMS10   EQU  FLASE
SMS20   EQU  TRUE
SMS30   EQU  FALSE
SMS40   EQU  FALSE
```

The same pattern applies for the setup of the 30, 40 MEG system

IO.ASM:

IO is the basic INPUT/OUTPUT routine used within the BIOS.

STDLST.ASM:

STDLST is the standard list device routine for the BIOS.

FLOP.ASM:

FLOP is the driver used for all the floppy drives.

PDRV.ASM:

PDRV is the driver for the PRIAM hard disk drive.

SADRV.ASM:

SADRV is the driver for the SHUGART 10 MEG drive.

QDRV.ASM:

QDRV is the driver for the SMS 10, 20, 30, 40 MEG drives.

BLK.ASM:

BLK is the BLOCKING/DEBLOCKING MULI SECTOR routine.

DATA.ASM:

DATA is the common data file which contains all of the necessary tables for the operation of the BIOS.

CBOOT.ASM:

CBOOT is the cold boot loader for all systems.

The following page is an example of the equate file for the STDSYS60 system.

```
;************************* SYSTEM SIZE ********************************
;
;       MSIZE IS THE EFFECTIVE SIZE OF YOUR SYSTEM AFTER ACCOUNTING
;       FOR CBIOS AND THE HOST BUFFERS. THE VALUE COMPUTED IS USED
;       AS THE NUMERIC VALUE 'NN' IN YOUR MOVCPM SEQUENCE:
;
;               E.G.    MOVCPM  NN *
;
;       MSIZE   EQU     60
;
;**********************************************************************
;       SYSTEM GENERATION EQUATES
;
;
TRUE    EQU     0FFFFH          ;DEFINE VALUE OF TRUE.
FALSE   EQU     NOT TRUE        ;DEFINE VALUE OF FALSE.
;
DPROM   EQU     TRUE            ;TRUE IF I.O. IS TO PROM
;
MMPM    EQU     FALSE           ;IF MP/M SET TRUE
;
MLDR    EQU     FALSE           ;IF M/PM LOADER SET TRUE
XMLDR   EQU     FALSE           ;IF M/PM LOADER W/O BOOT
;
DBLOCK  EQU     FALSE           ;TRUE FOR MULTISECTOR OPERATION
F256    EQU     FALSE           ;TRUE FOR 256 SECTOR FLOP
F512    EQU     FALSE           ;TRUE FOR 512 SECTOR FLOP
MAXHST  EQU     512D            ;SIZE OF BLOCKING BUFFER
;
SKEW    EQU     FALSE           ;TRUE IF HI SPEED FLOPPY
;
MAXI8   EQU     TRUE            ;TRUE IF STANDARD 8" FLOPPY
;
MINI48  EQU     FALSE           ;TRUE IF MINI DISK DRIVE 48TPI
MINI96  EQU     FALSE           ;TRUE IF MINI DISK DRIVE 96TPI
;
PRIAM   EQU     FALSE           ;TRUE IF PRIAM HARD DISK
;
SHUGART EQU     FALSE           ;TRUE IF SHUGART 1000 SERIES HD
;
QUANTUM EQU     FALSE           ;TRUE IF QUANTUM SERIES 2000 HD
SMS10   EQU     FALSE           ;TRUE IF SMS 10 MEG
SMS20   EQU     FALSE           ;TRUE IF SMS 20 MEG
SMS30   EQU     FALSE           ;TRUE IF SMS 30 MEG
SMS40   EQU     FALSE           ;TRUE IF SMS 40 MEG
;
SEKRATE EQU     01              ;00 = 3MS, |01 = 6MS, |02 = 10MS, |03 = 15MS
;                                QUME,850 |801       |            |
;                                MINIS    |          |            |
DUAL    EQU     FALSE           ;TRUE IF DUAL SPINDLE DRIVE (PERSCI)
;
DEBUG   EQU     FALSE           ;TRACE REPORTS ENABLED IF TRUE
;
SIDE2BIT EQU    FALSE           ;SIDE TWO COMPARE BIT (NORMALLY OFF)
```

2.12

MODIFYING and GENERATING CP/M OPERATING SYSTEMS

The CP/M operating system is composed of several parts: the BDOS (Basic Disk Operating System), the BIOS (Basic Input Output System), the CCP (Command Control Processor), and the BOOT (Cold Boot Loader).

In the US MICRO SALES system, the programs BOOT and BIOS are broken into several parts that are concatenated at the time that the system is built into one large module. The reason this is done is that fewer base modules need to be maintained to support all of the varied disk systems. Also, most larger multiuser modules would not fit in the system under the editor.

This BIOS system is intended to be used with SUBMIT modules that bear the name of the object BIOS produced. For example, PRISYS60.SUB contains the following:

```
PIP
PRISYS60.ASM=PRIEQU60.ASM,IO.ASM,FLOP.ASM,PDRV.ASM,BLK.ASM,DATA.A
SM,CBOOT.ASM
MACRO PBIOS
```

To concatenate this module, type: SUBMIT PRISYS60 <CR>

The PIP program picks out the necessary modules from the disk you have reserved for systems and creates the correct BIOS. The MACRO assembler will then assemble it for you.

Maintaining operating systems is a problem in that often minor "fixes" need to be added to them and having a dozen or so "versions" around means that not all of them will get the latest updates.

One of the greatest advantages of this system is that you only maintain one set of basic modules. In fact, we recommend that you keep only one backup copy of the disk containing these modules. In that way, you can be sure that the software you are building is the latest revision.

Modifications you wish to make should be added by putting conditional assembly flags around the code that is optional and establishing a flag in the main xxxEQU60.ASM header file. This keeps confusion and multiple copies to a minimum. Also, if you need to see if a certain group of files are present you can type: "DIR *.SUB" and just the system building files will appear. This is a real time saver when dealing with a hard disk that can display 4 or 5 screens of directory!

Programs you will need on the system disk to assemble a system:

1. ALL NECESSARY ... .ASM
2. CUSTOM BUILT ...SYS.SUB
3. SUBMIT.COM
4. MACRO.COM
5. MACRO.LIB
6. Z80.LIB
7. MOVCPM.COM
8. ED OR POLYVIEW (Polyview is HIGHLY recommended over ED)
9. DSYSGEN.COM

STEP BY STEP
A - MODIFICATION ***

The first thing to do is to edit the appropriate xxxEQU60.ASM and set all the equates to suit the configuration of the system you desire.

B - EXAMINATION ***

Next, type the ...SYS60.SUB file and look at the subparts needed to concatenate the final BIOS. Then Type DIR *.ASM and see if all the correct files are present.

C - CONCATENATION AND ASSEMBLY ***

If they are, type SUBMIT ...SYS60 The final product will be a ...SYS60.HEX file.

D - GENERATING CP/M ***

Type "MOVCPM 60 *"

The program MOVCPM.COM will create the BDOS for you. Usually, this needs to be done only once. Spend some time and make several of the common system sizes (e.g. 63K, 62K, 48K, etc.). When the MOVCPM has completed its job, type: "SAVE 34 CPMXX.COM" where XX is the system size.

E - PUTTING SYSTEM ON SYSTEM TRACKS ***

This is the job of DSYSGEN.COM. When you load DSYSGEN, it will ask you if the system has a file on disk. Answer 'N'. Then it will ask you if you want to build a system. This time, answer 'Y'. DSYSGEN will then ask you for the name of the CP/M file and then the name of the BIOS file. It will fetch the data that you have prepared and put it in the correct places in preparation for installation on the disk of your choice.

The next thing that will appear on the screen is a large table which has at the left of it some code bytes that will be written to the last byte of the first sector of the disk chosen.

At this time, the system you have created can be put on any of the disk formats specified in the table. Of course, the hardware must be compatible with the type of media you have picked and the disk must have been formatted with a format compatible with your choice (i.e. you could not put a 512 SKEW system on a 128 byte, double density disk.)

It should be noted that you need not put the system on a disk at this time. In fact, the computer system on which the operating system was created need not even be capable of writing to the drive type that the program you have built is designed for. You can type a control 'C' in answer to the "DESTINATION DRIVE" question and then type: "SAVE xx xxxCPMxx.COM" (the x's are unique system identifiers). The prepared operating system can be transported to the desired system via the floppy disk and read in using the first option on DSYSGEN (The one which asks if the system is a file on disk). For more information on how to use the DSYSGEN utility, refer to that section in this manual.

A third method for transferring operating systems is achieved by answering 'N' to both of the first questions DSYSGEN asks. DSYSGEN will then ask for the codebytes of the source disk you are going to take the operating system from. Then it will ask for the drive. From that point on you can proceed just as if you had constructed a system or read it as a file from disk.

The programmer must make sure that the operating system put on the system tracks will function properly. Very little error checking can be done in this process. It is extremely easy to make a "fatal" error. One of the most common is assembling a system that is too large to fit on the system tracks.

The best we can suggest for checking the load size is to go ahead and try to put the system on the disk. The DSYSGEN program will report to the screen the system size, load address, ending address and BIOS jump table adress for the current system. Also it will see if the system will fit on the system tracks of the media you have selected.

Use the following as a guide for how large systems may get:

PROGRAM	SIZE	TRACKS REQUIRED	IMAGE LOAD ADR
COLD BOOT	80H	1	900H - 97FH
CCP	800H	16	980H - 117FH
BDOS	E00H	28	1180H - 1F7FH
BIOS	F80H	31	1F80H - 2F00H
	2600H	76	900H - 2F00H

F - SYSTEM TRACKS AVAILABLE ON VARIOUS MEDIA ***

MEDIA TYPE	TRK-0	TRK-1	TRK-2	TOTAL
5" 128 DD	18	30	30	78
5" 512 DD	18	35	35	88
8" 128 SD	26	26	--	52
8" 128 DD	26	51	--	77
8" 512 DD	26	64	--	90
8" SHUGART HD	64	64	--	128
8" QUANTUM HD	128	--	--	128
14" PRIAM	104	--	--	104

Note that these values are relative to 128 byte blocks. That is to say, if you multiply the total on the right by 128, the result is what DSYSGEN will record on the system tracks.

If your system exceeds the 76 track guideline used in the example for using DDT to check your system size, you will have to adjust the upper boundary accordingly. Simply add the system size to the image base address and look with DDT at the resulting RAM location.

You also can dump the system tracks into RAM after writing them by re-entering DSYSGEN and letting it get the system by specifying the drive at the prompt "SOURCE DISK?". Exit and type: "SAVE 40 (40 may not be enough) xxCPMxx.COM". Remember, the 40 is PAGES so a SAVE 40 = 40 X 256 = 10,240 or 2800H. If your system is very large, you will have to do some math.

G - DESTINATION DISK PARAMETERS ***

After you specify a destination drive, a large menu will fill the screen. You will be prompted to choose the type of media that you wish to write the system to. The DSYSGEN program is quite universal in that it can service all the system types U S Micro supports. Be sure the computer system and media you are writing to are capable of supporting the operating system you are putting on them. No error checking will be done.

Occasionally, a message will appear that says "CODE BYTE FOR OPERATING SYSTEM AND MEDIA DO NOT AGREE, CONTINUE?" This is caused because the formatting program (DFOCO) has installed what it thinks is the correct code byte for this system in the last byte of the boot sector (7FH). DFOCO is usually correct. If you get this message, go over what you are doing once again, you probably will find something wrong. You may override the warning and write the operating system to the disk.

2.13

DSYSGEN OPERATING INSTRUCTIONS

DSYSGEN is a program designed to put C/PM, M/PM or DP/NET operating systems on the system tracks of a floppy or hard disk mass storage unit.

The C/PM operating system consists of a cold boot loader, the CCP (Command Control Processor), BDOS (Basic Disk Operating System) and the BIOS, (Basic Input - Output System).

On a floppy disk, the operating system resides on the first two tracks of the disk. The monitor PROM that signs on immediatly after a reset contains a short boot routine that reads the first sector into memory at 0000H and jumps to 0000. The cold boot loader, as it is called, then proceeds to read in all of C/PM and put it where it belongs in the system memory. The cold boot loader then jumps to the cold boot entry point in the BIOS and initializes the I/O and completes the sign on.

To prepare for putting the system on the disk, an image of the final operating system must be assembled in RAM. Two seperate programs must be prepared.

The first is the C/PM BDOS and CCP. It is prepared by using the MOVCPM utility on your C/PM distribution disk. When you are sure of the size of the system simply type: "MOVCPM <size> *". There will be a short delay and then a message will appear saying "Type SAVE 40 C/PMxx.COM" where xx will be the size.

The other program is a little more difficult in that you must correctly set some "equates" at the beginning to configure it to operate the hardware you are running. In the appendix, you will find some examples of single user, multi user, network and hard disk configurations that should be of help in determining the right way to set up the BIOS. You will notice that the BIOS is now broken into several subparts. The parts are explained as follows:

I/O = Cold boot loader, console input and output, common drive select routines, logical unit map, miscellaneous routines and messages needed by all configurations. I/O must be included in all system configuations.

STDLST = Printer driver, (There are several versions for commonly-used printers such as Centronics, Diablo, etc.)

FLOP = Floppy disk drivers.

PDRV = Priam hard disk drivers.

SADRV = Shugart and Quantum hard disk drivers.

BLK = Blocking and De-blocking drivers for use with sector sizes greater than 128.

DATA = Disk parameter tables, skew tables, data storage and disk drive buffers.

CBOOT = Cold boot sign on code that gets overwritten by the directory buffer and de-blocking buffer.

INTDRV = Interrupt driver routines (may be included when M/PM systems are built).

The parts of the BIOS are put together using the concatenation feature of PIP. To make the system simply choose the appropriate xxSYS.SUB file that assembles the configuration you need and type "SUBMIT xxSYS . The MACRO assembler is required along with MACRO.LIB and Z80.LIB to construct most modules.

The equates in IO should be modified using a text editor like the one supplied on your distribution disk called ED.COM. ED is difficult to use. We recommend, for assembly language work, that you purchase Polyview by MICRO CONCEPTS in Fullerton, Ca. Word Star by Micro Pro would be an acceptable substitute, but is not as easy to use as Polyview on assembly formats.

The size of the system can be adjusted by changing the MSIZE equate in the BIOS. This number should always be the same as the size you used to create the C/PM BDOS.

Once you have the CP/M image saved on disk and have modified the equates correctly to configure the BIOS to your hardware, run the SUBMIT program on the xxSYS.SUB which links together the SUB modules to create the final BIOS and assemble the whole thing.

At this point, we will assume you have both CPMxx.COM and xxSYS.HEX on the same disk with DSYSGEN.COM. Notice that there is no longer a separate BOOT.HEX file. It is included in the IO section of the BIOS. There is also no need to find the 'BIAS' values to tack onto the BIOS, the DSYSGEN program figures all of that out for you.

When DSYSGEN signs on it will ask if the system you wish to install on the drive exists as a xxxxx.SYS file on the disk already. If you are sure to use the current configuration again, it would be wise to save it after building it with DSYSGEN as a file. Then you need only answer 'Y' to "IS THE SYSTEM ON THE DISK". This saves a few steps. If you are supporting several configurations of mini and 8" drives, various printers, etc., you could keep one diskette with DSYSGEN and all the system files on it. Then putting them on the drives is a snap.

For the moment, we will assume the system has to be built. Answer 'Y' to the query: "DO YOU WANT TO CONSTRUCT A SYSTEM". DSYSGEN will ask for the drive and file name for both C/PM and what is called BOOT + BIOS. DSYSGEN will ask for a destination drive, confirm it, then display a rather large menu.

This menu is a guide for the code bytes stored in the last byte of the first sector of all drives. The appendix explains the meaning of each bit in the code byte, if you are interested. Other details are included for your information. This data was taken from a listing of DSYSGEN, so it is in rather raw form.

Whenever possible, it is advisable to choose the "SKEW" option. Many companies are currently raving about their 'four times faster' C/PM. Optimizing the skew factor is all that is required to multiply disk performance (You should try the double-sided 512 byte 8" configuration if you would like to see a floppy perform like a hard disk).

Using the DFOCO (for Double density FOrmat and COpy) utility, you should have already prepared your disk with the correct format. After answering with the correct code byte, DSYSGEN will write the system onto the disk. If the code byte on the disk and the code byte you have chosen do not agree, the following message will appear: "The new system type and the destination disk format do not match. Do you want to continue (Y or N) ?". This may or may not be ok. DFOCO does not know what you will use for a code byte and has some fixed ideas about what it puts on the disks. Situations such as putting a code byte for an 8" floppy disk on a hard disk drive are incorrect, but choosing a skew code when DFOCO has not put one there (or vice versa) is ok. Presently, skew codes are automatically put on 512 byte sectored disks by DFOCO because we assume you are after the speed and the extra space which that format provides. On other formats such as 128 byte, double density, skew is not assumed because you may have compatability problems with existing disks. In any case, DSYSGEN will overwrite whatever code byte is there if you tell it to.

After writing the system out, DSYSGEN will state that the function is complete and ask if you want to do another disk or hit 'return' and re-boot. Upon being re-booted, it will display a "SAVE" message with a size that has been calculated for the number of bytes in your current system size. If you use this option, you can avoid going through the whole system building process again by simply using the image off disk.

Some attempts are made in the DSYSGEN program to aid you in building a system that will not overwrap FFFF and destroy the boot program that is loading the system. If DSYSGEN detects that your system is too large for the memory size you have chosen, it will tell you so. If the system will not fit on the system tracks of the disk you are using it will notify you of this, as well. In the latter case, DSYSGEN will trim the system to fit (not doing so would hang the disk controller).

2.14 DSYSGEN - STEP-BY-STEP

After formatting a blank diskette using the DFOCO program, the following procedure will accomplish the assembly and transfer of an operating system to tracks 0 and 1 of the diskette:

1. Type "DSYSGEN" and carriage return (<CR>).

2. The program will load and ask, "System source file on disk? (Y/N)". The answer to this question will always be "N" unless you have saved the system you wish to use from a previous DSYSGEN operation (See last step in this procedure).

3. Upon answering "N", the program will respond with the question,"Construct file? (Y/N)". If you wish to go no further in this procedure for some reason, answer "N". Otherwise, the answer will always be "Y".

4. After a "Y" answer to the preceding question, the program will instruct you to enter "Drive and name of CPMxx.COM" and then display this: "DRIVE...". The program is asking for the DRIVE on which the CPMxx.COM file is located. This file is found on the "A" diskette which you received with your system, however, that "A" diskette may be in your "B" drive, for example, or your "C" drive or your "G" drive. The point is that the program is asking for the DRIVE designator, not the diskette designator. After typing the correct designator, type a <CR>.

5. The program now asks for the file name. In all XOR systems now in production the file name is: "CPM60.COM" followed by a <CR>.

6. At this point, the program will state "Reading file..." and then respond with the statement "Drive and name of boot + BIOS.HEX" and ask again "DRIVE...". The boot + BIOS.HEX file is located on your "B" DISKETTE, therefore, remove the "A" diskette from drive A and put in the "B" diskette. Now, answer the question "DRIVE..." with "A". The program will respond with "File name...". For the XOR S-100-12, two files may be used: STDSYS60.HEX or BLKSYS60.HEX. STDSYS60.HEX is capable of reading and writing only diskettes formatted with 128 Byte sectors. BLKSYS60.HEX is capable of reading and writing 512 Byte sectored diskettes as well as those sectored at 128 Bytes (256 Byte sectored diskettes may be read and written to by setting the proper equate true in the BLKEQU60.ASM file before beginning the DSYSGEN procedure). The choice is yours.

7. After entering the boot + BIOS.HEX file name, the program will again display "Reading file..." and then display a long menu. At the bottom of this menu will be the statement "Enter code bytes for destination disk -->". The leftmost column of the menu contains the code Bytes. To the right of each code Byte is a shorthand description of the meaning of each code Byte. For example: Code Byte 91/99. The description to the right reads: 2D/256 CPM 8" Std/Skew DOUBLE sided mode. This means that the operating system you create with this Byte is for double sided, double density 8" diskettes sectored at 256 Bytes per sector. It is a CP/M operating system which will read and write to a standard formatted diskette (Code Byte 91) or a diskette formatted with skew (Code Byte 99) which is faster.

8. After entering the code Byte, the program will display some information about the operating system you have just created (i.e. the size of the system, the address at which it will load, the last address under which it will load, and the BIOS jump table address). It will then ask "Destination drive? (or <CR> to re-boot and save image)". At this time you may put the operating system on the diskette you have formatted (by typing in the DRIVE designator for the drive containing that diskette) or you may save the operating system on the diskette currently in your "A" drive (by typing a <CR>). If you choose the former option, after typing in the drive designator the program will respond with "Funtion complete (<CR> to continue or <^C> to re-boot and save)". If you type a <CR>, the program will go back to the code Byte menu to allow you to put your system on another diskette. If you type a ^C, the program will respond with "To save system type, 'Save ## xxCPM<size>.SYS (where xx = reference indentifiers)". The symbol "##" will be a number telling you the size of the block to save, size is also given to you numerically. The "xx" are indentifiers which you choose to uniquely indetify this particular operating system for future use. Should you answer the Destination drive question with a <CR>, the system will immediately go to the "To save system..." statement.

9. If you save the system on diskette, you will be able to answer the first question of this procedure ("System source file on disk?) with a "Y" and proceed directly to Step #7.

MPM II SYSTEM GENERATION PROCEDURE

REBUILDING A SYSTEM

Generation of an MP/M operating system is a simple procedure. It cannot be overstated, however, that these large, highly integrated, multiuser operating systems are a bit tempermental. They can be modified, but unless you have a great deal of experience or a great deal of time, don't try it! Once an operating system has been modified, getting it to run may be an arduous chore. Common errors such as generating modules that are too large, moving interrupt handler page aligned bounderies, and changing code that should not be changed are easily made but can be a nightmare to unravel.

The most common reason for modifying the operating system is to install custom printer drivers. The system component file (MSTDLST.ASM) is a self-contained unit that can be modified without a complete understanding of MP/M. Testing your printer drivers can be done in a single user mode by writing a small driver that calls this module. See your MP/M II manual for entry conventions.

Our support for modified operating systems is limited. If you run into problems, our response will be to ask you to submit your problems to us on a diskette accompanied by a written explaination. One of our engineers will then examine them using our hardware/software debugging tools.

Regenerating an operating system can be accomplished by typing "SUBMIT PMPM2SYS" (for a Priam system). As long as you are not changing any system parameters, the computer will complete the regeneration process automatically. To do this, both of the distribution diskettes, received with your system, will have to be copied to drive 'A' of your system using the "PIP" program. If your system contains a hard disk drive, this will be no problem. If drive 'A' is a floppy disk drive, you may have to use double-sided diskettes formatted with 512 byte sectors in order to be able to copy both distribution diskettes onto one master diskette.

The GENSYS program gathers many separate pre-assembled modules and combines them with immediate input from the keyboard to produce the run time operating system (MPM.SYS).

MPM.SYS does not reside on disk track 0 as in CP/M. You will find it as just another file on the disk. On "system" track 0 is a program called MPMLDR. MPMLDR.COM must be brought into RAM at 100h. It is about 2000h long. It will look on the disk for MPM.SYS, load the system parts and report load addresses to the screen. MPMLOADER, therefore, should be thought of exactly like a conventional CP/M system, as far as DSYSGEN is concerned. If you want to run a system under CP/M, you put BOOT, BDOS, and BIOS out on the system tracks using the DSYSGEN utility. If you want the system to boot MP/M (or CP/NET) you must put MPMLDR.COM on the system tracks. MPMLDR consists of Digital Research's MPMLDR.COM program and a shortened version of the BIOS that has been trimmed of all disk write and other unecessary routines.

MPMLDR rarely has to be changed. MPM.SYS, which is composed of XIOS, XDOS, BDOS, SYSTEM DATA PAGE, and any RSP-type programs, is what usually gets changed.

We have broken the explaination of the generation of the operating system into two parts: 1) GENSYS - where all the modules are melded together and 2) REGEN - where the XIOS.SPR is completely rebuilt.

/GENSYS/

The following example assumes a XOR Products 2.0 MP/M system generation involving a Priam drive and 4 user banks of memory (In our system, the system bank is set up with a base of 100h so that only short relocateable PRL type files will load under the operating system. This resolves the problem of one bank being short of memory for many programs that have been known to "crash" the operating system when running in bank 0.).

Note: Under MP/M II, it is possible to generate an operating system using the most recently set values in the SYSTEM.DAT file by typing in $A after GENSYS. It is advisable to put a printer on line during your first attempt and to allow the pre-assigned values we have set to be assigned on the first pass. Then after examination of each, cautious modification can be made.

The following is an example of what will appear:

```
MP/M II V2.0 System Generation
Copyright (C) 1981, Digital Research
Default entries are shown in (parens).
Default base is Hex, precede entry with # for decimal

Use SYSTEM.DAT for defaults (Y)?
Top page of operating system (FF)?
Number of TMPs (system consoles) (#5)?
Number of Printers (#5)?
Breakpoint RST (07)?
```

(cont'd.)

```
Add system call user stacks (N)?
Z80 CPU (Y)?
Number of ticks/second (#60)?
System Drive (A:)?
Temporary file drive (A:)?
Maximum locked records/process (#16)?
Total locked records/system (#32)?
Maximum open files/process (#16)?
Total open files/system (#32)?
Bank switched memory (Y)?
Number of user memory segments (#5)?
Common memory base page (C0)?
Dayfile logging at console (N)?

SYSTEM   DAT   FF00H   0100H
TMPD     DAT   FD00H   0200H
XIOSJMP  TBL   FC00H   0100H

Accept new system data page entries (Y)?
RESBDOS  SPR   F000H   0C00H
XDOS     SPR   CE00H   2200H

Select Resident and Banked System Processes:
ABORT    RSP   (N) ?
MPMSTAT  RSP   (N) ?
SCHED    RSP   (N) ?
SPOOL    RSP   (N) ?

BNKXIOS  SPR   B700H   1700H
BNKBDOS  SPR   9400H   2300H
BNKXDOS  SPR   9200H   0200H
TMP      SPR   8E00H   0400H

LCKLSTS  DAT   8B00H   0300H
CONSOLE  DAT   8600H   0500H

Enter memory segment table:
Base,size,attrib,bank   (86,7A,80,00) ?
Base,size,attrib,bank   (00,C0,00,01) ?
Base,size,attrib,bank   (00,C0,00,02) ?
Base,size,attrib,bank   (00,C0,00,03) ?
Base,size,attrib,bank   (00,C0,00,04) ?
Base,size,attrib,bank   (02,80,00,00) ?

MP/M II  Sys   8600H   7A00H   Bank 00
Memseg   Usr   0000H   C000H   Bank 01
Memseg   Usr   0000H   C000H   Bank 02
Memseg   Usr   0000H   C000H   Bank 03
Memseg   Usr   0000H   C000H   Bank 04
Memseg   Usr   0200H   8000H   Bank 00

Accept new memory segment table entries (Y)?

** GENSYS DONE **
```

The GENSYS program will now build an MPM.SYS that will include the XIOS.SPR (that resides on the currently logged disk), BNKBDOS.SPR, XDOS.SPR, and BDOS.SPR.

MPMLDR.COM should be run now to see if any of the changes in the operating system have pushed parts of the system that must remain in common memory beyond their boundry (1F00H).

To run the MP/M operating system that you have generated, type:

```
A>MPMLDR <CR>
```

System response:

```
MP/M Loader
===========

Number of consoles  = 5
Breakpoint RST #    = 7
Z-80 CPU
Top of memory       = FFFFH

Memory Segment Table:
SYSTEM     DAT  FF00H     0100H
CONSOLE    DAT  F900H     0500H
XIOS       SPR  E600H     1300H
BDOS       SPR  DE00H     0800H
XDOS       SPR  C000H     1F00H      <---- Everything above must
BNKBDOS    SPR  B200H     0E00H            be in common memory
-------------------------------
Memseg     Usr  0100H     B000H           ; Bank 0 (no COM files)
Memseg     Usr  0000H     C000H           ; Bank 1
Memseg     Usr  0000H     C000H           ; Bank 2
Memseg     Usr  0000H     C000H           ; Bank 3
Memseg     Usr  0000H     C000H           ; Bank 4
```

For a more in depth explanation of all the system parameters, consult your MP/M II manual.

/REGEN XIOS/

The BNKXIOS.ASM is on the disk in several parts. The components are similar to those used in building single-user CP/M systems. One additional component is the interrupt handlers program (INTDRV.ASM). Other programs contain the disk controller drivers for the 8" or 14" hard disk drive controllers. (i.e. PDRV.ASM for PRIAM and SADRV.ASM for SHUGART) The ASM files for the I/O or disk drivers may be modified with the ED or POLYVIEW programs. You may assemble the modules and generate a complete system quite simply by typing SUBMIT PMPM2SYS (for a Priam system). The submit program of the appropriate type for the disk drivers you are employing will concatenate the input handlers and the disk drivers, go through the necessary steps to produce the final .SPR file, and even do the GENSYS for you.

There are currently four MPM configuations being supported:

```
PMPMSYS =       Priam 14" 33.5, 67, 150 MEGABYTE
DMPMSYS =       Shugart 8" 10 MEGABYTE
SMPMSYS =       Quantum 8" 20 & 40 MEGABYTE
FMPMSYS =       Floppy 1 & 2 megabyte 2 user
```

A thorough understanding of the meaning of each of the lines in the various equate headers is essential for successful system re-configuration.

As was stated earlier it would be advisable to let the automatic generation proceed once with a printer on line. Then GENSYS could be run again to modify system parameters after you have become familiar with the reasons for the choices already made.

When the RMAC (relocateable assembler) is working on the file, some intentional messages will appear on the screen.

The + ENDM
+ ENDM
+ ENDM should be ignored. It is the product of the automatic generation of interrupt handlers.

The " U(ADDR)3E00 MVI A,ERROR ; [comment]" messages are self explanatory in that certain boundaries in the interrupt handlers cannot overlap. ERROR is intentionally not defined and the addresses of these areas can be checked before run time by hitting a control S to stop the screen. The 512 blocking-deblocking buffer overwrites the initialization routines and care must be taken that added code does not limit this area.

/MPMLDR.COM/

The MPMLDR.COM does not need to be modified. When called it will search for the MPM.SYS program and load it.

To get the MPMLDR.COM on to the system tracks, you might want to read the procedures in the section on DSYSGEN for CP/M and assume the MPMLDR is the same as any other system you would like to have DSYSGEN write on the system tracks. Think of DSYSGEN as a general purpose program that will accept any data that is in the correct form and write it to the disk. If it happens to be a CP/M system, then the system will boot CP/M. If, on the other hand, it is MPMLDR, then the system will "come up" MP/M. The main difference between the two is in the boot programs that occupy the first 128 bytes of the system area (see memory map).

The CP/M boot program comes in at location 0000h and continues to read sector 2 through ?? at the DMA address pointed to at the end of the boot program. (see boot source for NSECTS and DMADD EQU's in the system component program IO.ASM.)

On the other hand, the boot routine for MP/M loads MPMLDR.COM at 100H and uses its own BIOS at 1700H to open the MPM.SYS file on the 'A' drive of that BIOS and load it to system memory.

The MPMLDR.COM that is on your distribution disk runs at 100H and has been made up of a special short BIOS that is org'd at 1700H, and the proprietary Digtal Reasearch MPMLDR.COM. DRI's MPMLDR occupies the space from 100H to 16FFH and our BIOS extends above that to whatever length is necessary.

To get this in a form that DSYSGEN will be able to put on the system tracks, we must use DDT to make a version that is offset by 880H. Do this by using the following example:

```
DDT <CR>
IMPMLDR.COM <CR>
R880 <CR>
^C   <RE-BOOT>
SAVE 32 XMPMLDR.COM          ("X" used to distinguish from unbiased)
```

Remember that while we are refering to MPMLDR we mean that it has been properly mixed with the supplied Digital Research driver that runs at 100H and our own BIOS we have ORG'ed at 1700H and saved the combination calling it MPMLDR.COM. The version supplied by DRI won't run without our BIOS.

While the MPMLDR.COM we have supplied has its own BIOS that would work in the final version, it does not have a cold boot section at the front. You have to generate another BIOS in the following manner:

To complete the construction of a bootable MPMLDR verify that the equate in the file PLDREQU.ASM (for a Priam Hard Disk) named MLDR has been set TRUE and the one named XMLDR is set FALSE.

Proceed by typing SUBMIT PMPMLDR. A HEX file will result that has the boot module attached at the front (of the hex file). Use the DSYSGEN program to put the system on the disk just as you would for a CP/M operating system.

EXAMPLE:

```
DSYSGEN <CR>
SYSTEM ON DISK AS FILE ?   (N)
CREATE SYSTEM ?            (Y)
DRIVE ?                    (A)
xxCPM.COM ?                (XMPMLDR.COM)
xxBOOT + BIOS.HEX ?        (PMPMLDR.HEX)
```

Complete the procedure by answering drive type and destination questions as appropriate.

```
*****************************************************************
**                                                             **
**                MEANING OF BITS IN CODE BYTE                 **
**                                                             **
*****************************************************************

                        ----------------------- ; types of drives
                        I     ----------------- ; sector sizes  and hard
                        I     I                   disk types
                        V     V
                       ____  ____
DBLDENS EQU            0001$0000B                ; double density flag bit
HARDISK EQU            0010$0000B                ; hard disk flag bit
MINI    EQU            0100$0000B                ; mini flag bit
DBLSID  EQU            1000$0000B                ; disk is double sided
SKEW    EQU            0000$1000B                ; sectors are skewed
128     EQU            0000$0000B                ; 128 bytes per sector
256     EQU            0000$0001B                ; 256 bytes per sector
512     EQU            0000$0010B                ; 512 bytes per sector
1024    EQU            0000$0011B                ; 1024 bytes per sector

 DENSITY FLAGS

                        I------------------------ DENSITY CODE
                        I     I------------------ SECTORS TO WRITE
                        I     I  I--------------- SECTORS PER TRACK TK 0 + 1
                        I     I  I  I------------ SECTORS PER TRACK TK 1 + 1
                        V     V  V  V  V--------- SECTORS MODULO 128
DENFLG:
        DB              050H,77,17,30,77          ; 48tpi ssdd 128
        DB              052H,77,17,30,77          ; 48tpi ssdd 512
        DB              0D0H,77,17,30,77          ; 48tpi dsdd 128
        DB              0D2H,77,17,30,77          ; 48tpi dsdd 512
        DB              054H,77,17,30,77          ; 96tpi ssdd 128
        DB              0D4H,77,17,30,77          ; 96tpi dsdd 128
        DB              056H,77,17,30,77          ; 96tpi ssdd 512
        DB              0D6H,77,17,30,77          ; 96tpi dsdd 512
        DB              020H,77,00,00,76          ; 5' hard disk
        DB              010H,76,25,51,76          ; std 2.2 ss
        DB              0DDH,76,25,51,76          ; std 1.4 & 2.2 ss
        DB              090H,76,25,51,76          ; std 2.2 ds
        DB              011H,51,25,26,77          ; 8" 256
        DB              019H,51,25,26,77          ; 8" 256
        DB              091H,51,25,26,77          ; 8" 256 ds
        DB              01AH,41,25,16,89          ; 8" 512 skew
        DB              09AH,41,25,16,89          ; 8" 512 skew
        DB              028H,77,103,103,103       ; PRIAM HARD DISK
        DB              02AH,55,00,00,55          ; Priam 512
        DB              025H,52,26,26,104         ; Shugart 1004 256
        DB              02EH,19,16,16,19          ; Quantum 2000 series
        DB              000H,51,25,26,51          ; default single density
```

```
DB      ' 50 - 2D/128 CPM  5" 48TPI mini single sided mode',0DH,0AH
DB      ' D0 - 2D/128 CPM  5" 48TPI mini DOUBLE sided mode',0DH,0AH
DB      ' 54 - 2D/128 CPM  5" 96TPI mini single sided mode',0DH,0AH
DB      ' D4 - 2D/128 CPM  5" 96TPI mini DOUBLE sided mode',0DH,0AH
DB      ' 52/5A 2D/512 CPM  5" 48TPI mini Std/Skw single sided mode',0DH,0AH
DB      ' D2/DA 2D/512 CPM  5" 48TPI mini Std/Skw DOUBLE sided mode',0DH,0AH
DB      ' 56/5E 2D/512 CPM  5" 96TPI mini Std/Skw single sided mode',0DH,0AH
DB      ' D6/DE 2D/512 CPM  5" 96TPI mini Std/Skw DOUBLE sided mode',0DH,0AH
DB      ' 20 - 512 CPM 2.2 5" Hard disk mode',0DH,0AH,0DH,0AH
DB      ' 10 - 2D/128 CPM  8" 1.4 & 2.2 compatible directory',0DH,0AH
DB      ' 90 - 2D/128 CPM  8" Std DOUBLE sided mode',0DH,0AH
DB      ' 11/19 2D/256 CPM  8" Std/Skew single sided mode',0DH,0AH
DB      ' 91/99 2D/256 CPM  8" Std/Skew DOUBLE sided mode',0DH,0AH
DB      ' 12/1A 2D/512 CPM  8" Std/Skew single sided mode',0DH,0AH
DB      ' 92/9A 2D/512 CPM  8" Std/Skew DOUBLE sided mode',0DH,0AH,0DH,0AH
DB      ' 28 - 128 CPM 2.2 Priam Hard disk mode',0dh,0ah
DB      ' 2A - 512 CPM 2.2 Priam Hard disk mode',0dh,0ah
DB      ' 25 - 256 CPM 2.2 Shugart Hard disk mode',0dh,0ah
DB      ' 2E - 512 CPM 2.2 Quantum Hard disk mode',0dh,0ah,0dh,0ah
```

2.17

```
**********************************************************************
**                  'SUB' TYPE FILES FOR SYSTEM BUILDING            **
**                (MP/M  System Without Hard Disk Drivers)          **
**********************************************************************
```

NOTE:

The following files are used to 'link' together various xxx.ASM file types to build single user, multi-user, multi-sector and (?) operating systems.

In our principal development system all single user source is kept on drive 'A', multiuser on drive 'B', and all DOC files and utilities on drive 'C'. The drive assignments we have shown here can, of course, be changed to suit your environment. It may, however, be of some signifcance to know why the assignments are as they are. It is also necessary to have certain support files on the correct drive. (i.e. MACRO. LIB and ?) Simulating the conditions that were known to work can be of some assistance in case of difficulty.

FX.SUB - Builds XIOS for M/PM with no hard disk drivers.

```
;
EQUATES:
                    INTRDRV                              IO
                    -------                              --
          MPMIO   =     TRUE          DPROM    =        TRUE
          TRACE   =     FALSE         MMPM     =        TRUE
          BANKSW  =     TRUE          DBLOCK   =        (TRUE/FALSE)
          HARDISK =     FALSE         MAXHST   =        ( 0,256,512)
          DSKINTR =     TRUE          MAXI8    =        TRUE
          V256    =     FALSE         MINI48   =        (TRUE/FALSE)
          MAXBANK =     3             MINI96   =        (TRUE/FALSE)
          MAXUSER =     2             PRIAM    =        FALSE
          BIT7    =     TRUE          SHUGART  =        FALSE
          SWITCH  =     0BH           QUANTUM  =        FALSE
          MPM     =     TRUE          DUAL     =        FALSE
          TYPEAH  =     32            DEBUG    =        FALSE
          DISKTIME=     600
;
**********************************************************************
A:PIP B:XIOS.ASM=B:INTDRV.ASM,A:IO.ASM,FLOP.ASM,DATA.ASM
A:SUBMIT G B:XIOS
**********************************************************************
;
PARAMETERS:

DRIVE A:          ; SUBMIT.COM, XSUB.COM, FX.SUB, G.SUB, GENMOD.COM,

DRIVE B:          ; INTRDRV.ASM, MACRO.COM, MACRO.LIB, Z80.LIB
```

```
**********************************************************************
**              'SUB' TYPE FILES FOR SYSTEM BUILDING                **
**            (8 or 5" Floppy System, 128 Byte Sectors)             **
**********************************************************************
```

NOTE:

The following files are used to 'link' together various xxx.ASM file types to build single user, multi-user, multi-sector and (?) operating systems.

In our principal development system all single user source is kept on drive 'A', multiuser on drive 'B', and all DOC files and utilities on drive 'C'. The drive assignments we have shown here can, of course, be changed to suit your environment. It may, however, be of some signifcance to know why the assignments are as they are. It is also necessary to have certain support files on the correct drive. (i.e. MACRO.LIB and ?) Simulating the conditions that were known to work can be of some assistance in case of difficulty.

```
MSYS.SUB - Builds 8" or 5" floppy system, 128 byte sectoring
;
EQUATES:
                  IO
                 ----
          DPROM  =       TRUE
          MMPM   =       FALSE
          DBLOCK =       FALSE
          MAXHST =       0
          MAXI8  =       TRUE
          MINI48 =       (TRUE/FALSE)
          MINI96 =       (TRUE/FALSE)
          PRIAM  =       FALSE
          SHUGART =      FALSE
          QUANTUM =      FALSE
          DUAL   =       FALSE
          DEBUG  =       FALSE
;
**********************************************************************
PIP MSYS.ASM=IO.ASM,FLOP.ASM,DATA.ASM,CBOOT.ASM
MACRO MSYS
**********************************************************************
;
PARAMETERS:

DRIVE A:          ; SUBMIT.COM, IO.SUB, FLOP.ASM, DATA.ASM, CBOOT.ASM,
                  ; MACRO.COM, MACRO.LIB, Z80.LIB
```

```
**********************************************************************
**              'SUB' TYPE FILES FOR SYSTEM BUILDING                **
**              (CP/M System With Hard Disk Drivers)                **
**********************************************************************
```

NOTE:

The following files are used to 'link' together various xxx.asm file types to build single user, multi-user, multi-sector and (?) operating systems.

In our principal development system all single user source is kept on drive 'A', multiuser on drive 'B', and all DOC files and utilities on drive 'C'. The drive assignments we have shown here can, of course, be changed to suit your environment. It may, however, be of some signifcance to know why the assignments are as they are. It is also necessary to have certain support files on the correct drive. (i.e. MACRO.LIB and ?) Simulating the conditions that were known to work can be of some assistance in case of difficulty.

Note that standard list device drivers (STDLST.ASM) are shown, but CENTLST.ASM, (Centronics) DIABLST.ASM, (Diablo) etc. may be substituted.

SASYS.SUB - Builds BIOS for C/PM with SHUGART hard disk drivers.

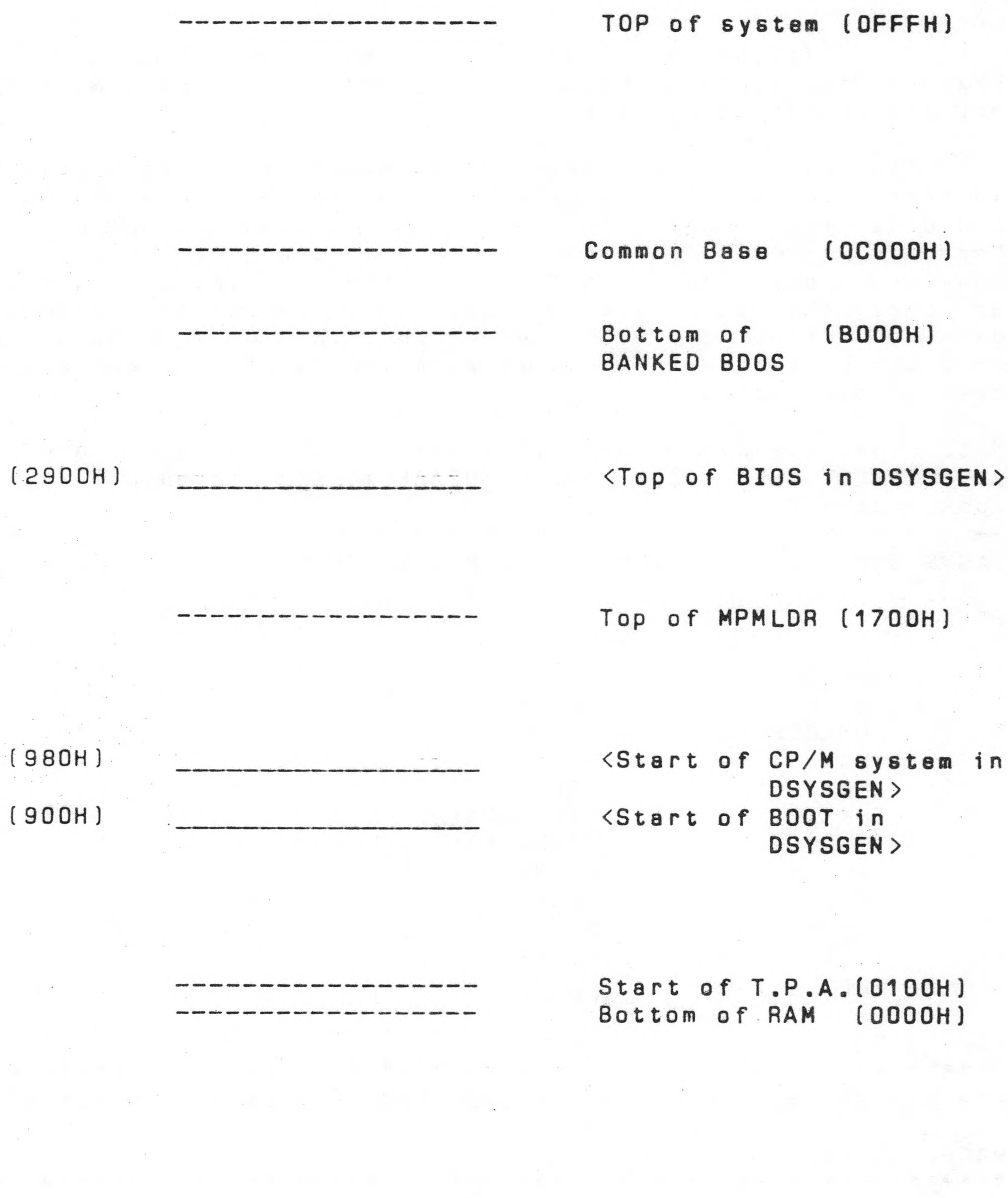
```
;
EQUATES:            IO
                    ----
                DPROM   =       TRUE
                MMPM    =       FALSE
                DBLOCK  =       TRUE
                MAXHST  =       ( 256 )
                MAXI8   =       TRUE
                MINI48  =       (TRUE/FALSE)
                MINI96  =       (TRUE/FALSE)
                PRIAM   =       FALSE
                SHUGART =       TRUE
                QUANTUM =       FALSE
                DUAL    =       FALSE
                DEBUG   =       FALSE

;SASYS.SUB
**********************************************************************
PIP A:SASYS.ASM=IO.ASM,STDLST.ASM,FLOP.ASM,SADRV.ASM,BLK.ASM,DATA.ASM,
                CBOOT.ASM
MACRO SASYS
**********************************************************************
;
PARAMETERS:

DRIVE A:            ; SUBMIT.COM, SASYS.SUB, MACRO.COM, MACRO.LIB, Z80.LIB
                    ; IO.ASM,STDLST.ASM,FLOP.ASM,SADRV.ASM,BLK.ASM,DATA.ASM
                    ; CBOOT.ASM
```

2.18

MEMORY MAP



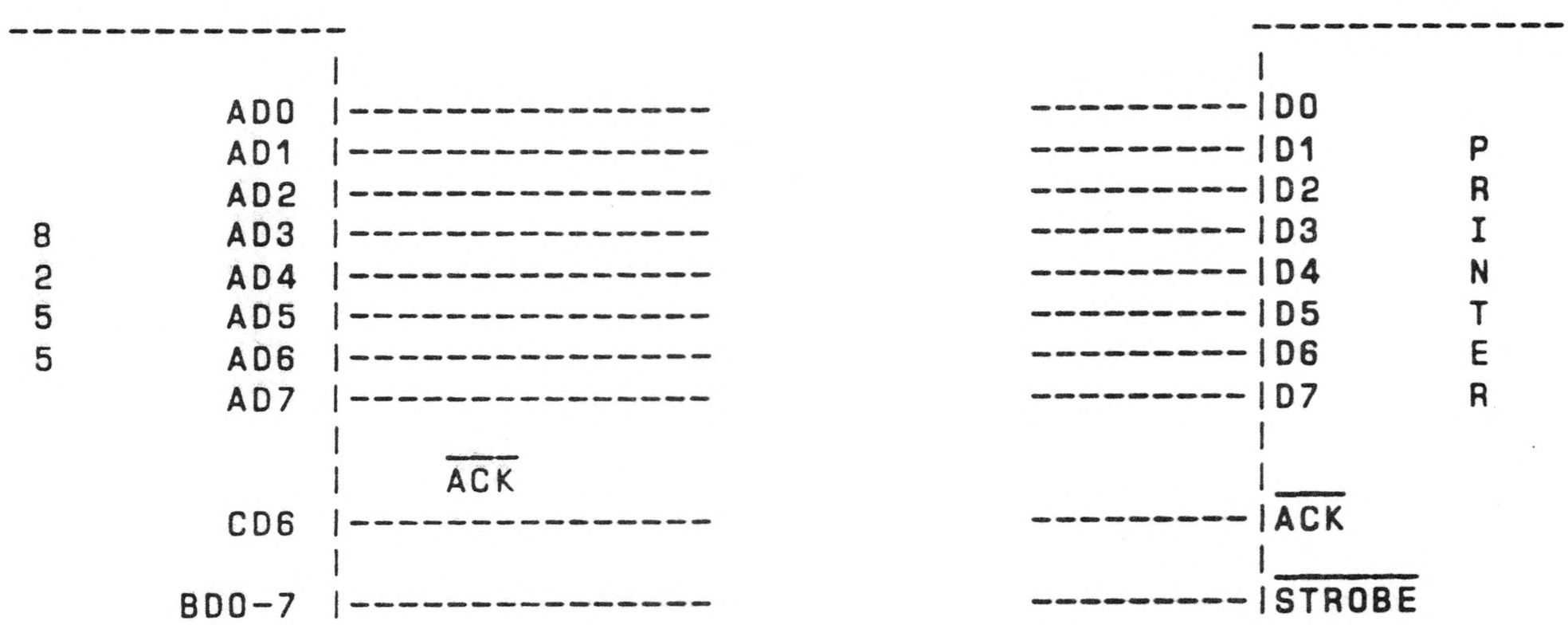
MICRO NOTE #1

PRINTER DRIVER-CENTRONICS INTERFACE

The code shown below can be installed into the BIOS by editing STDLST.ASM file and by replacing the code in that file with the code for the driver you wish to use in your system.

Printer drivers for the Centronics interface

This program is used for Centronics interface printers. It uses a 8255, PA0-7 are used for data transfer, PB0-7 are tied together for data strobe PC6 for ACK from printer.




```
LIST:
        LDA   INITFLG     ;
        CMA               ;
        JZ    LOOP        ;
        STA   INITFLG     ;
        MVI   A,0ACH      ;
        OUT   7           ; cmd port
        MVI   A,0FFH      ;
        OUT   5           ; B set hi

                          ;Centronics interface driver
LOOP:
        IN    06H         ; Check status of port
        ANI   0B0H        ;
        CPI   90H         ;
        JNZ   LOOP        ; if not ready check again
        MOV   A,C         ;
        OUT   04H         ; port 04 is data port
        MVI   A,00H       ; these toggle the strobe
        OUT   05H         ; line for the Centronics
        MVI   A,0FFH      ; printers
        OUT   05H         ;
        RET               ;

INITFLG:
        DB    0           ;
```

MICRO NOTE #2

PRINTER DRIVERS FOR DIABLO SERIAL PRINTER

On the interface paddle card "A" be sure to have all jumpers connected on the card (DTR, DSR, CTS, RTS). In this program an ETX is sent every 79 decimal characters then the system waits for an ACK from the DIABLO which responds with an ACK every time it reads an ETX.

```
;Diablo printer driver
LST:
        IN      01H             ;CHECK STAT
        ANI     02H             ;RECEIVE BIT ON?
        JZ      CHKBUF          ;NO DATA READY
        IN      00H             ;SOME DATA!!
        ANI     7FH             ;MASK PARITY
        CPI     06H             ;ACK?
        JZ      RSTCNT          ;RESET COUNT THEIR BUFFER EMPTY
        CPI     20H             ;NEW LISTING
        JZ      NEWLIST         ;CLEAR COUNTER

CHKBUF:
        LDA     STOR            ;CHAR SENT
        CPI     4EH             ;1 LESS THAN MAX
        JZ      ETXOUT          ;SEND FIRST ETX FOR FIRST OF 2 79 BYTE SET
        CPI     0CEH            ;TOP OF BUFFER
        JZ      ETXOUT1         ;ANOTHER ETX
        CPI     0CFH            ;TOP OF BUFFER
        JZ      LST             ;OVER LIMIT WAIT FOR ACK

OUTDAT:
        IN      01H             ;CHECK STAT
        ANI     01H
        JZ      OUTDAT          ;STAT OK?
        MOV     A,C
OUTDAT1:
        OUT     00H
        LDA     STOR
        INR     A               ;INCREMENT BUFFER COUNT
        STA     STOR
        RET

ETXOUT:
        MVI     A,80H           ;FIRST CLR OF BUFFER COUNTER USED ONCE
        STA     STOR            ;AT START ONLY OR NEWLST
ETXOUT1:
        IN      01H             ;CHECK STAT
        ANI     01H
        JZ      ETXOUT1
        MVI     A,03H           ;SET A FOR ETX OUTPUT
        CALL    OUTDAT1
        JMP     LST             ;JMP LST TO WAIT FOR ACK
```

```
NEWLIST:
        XRA     A                       ;CLEAR A TO 00H FOR NEW COUNT
        STA     STOR
        JMP     OUTDAT

RSTCNT:
        MVI     A,80H                   ;CLR BUFFER COUNT FOR COUNT OF 80H TO CFH
        STA     STOR
        JMP     LST

STOR:   DB 00
```

MICRO NOTE #3

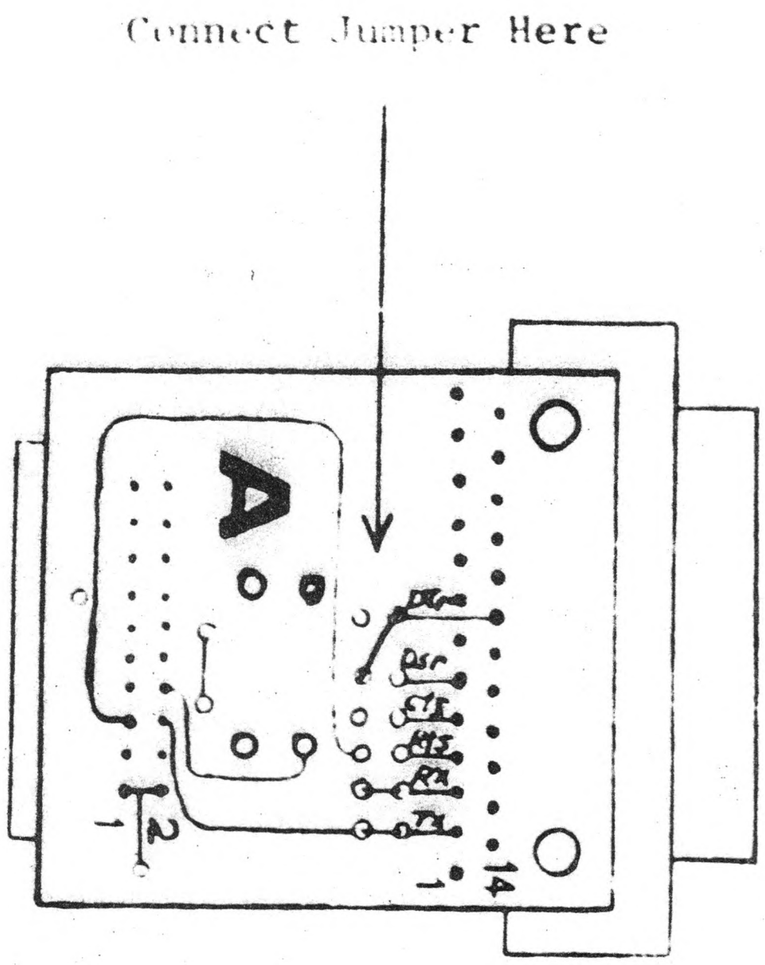
EPSON MX-80-SERIAL INTERFACE

Install the serial interface adapter in the Epson MX-80 and set all DIP switches on the serial board to the "off" position. This will select 9600 Baud, 8 bit word length, and parity disabled.

To set the baud rate for the 'A' port on the XOR CPU board to 9600, switch the top four switches identically to the bottom four switches on the 8 position DIP switch resient on the CPU board, (Port 'B' has been set at 9600 for terminal operation).

Finally, the 'A' board at the end of the serial cable must have one jumper installed for handshaking to take place. Please refer to the diagram below and install a jumper from DTR (printer side) to DSR (CPU) side).

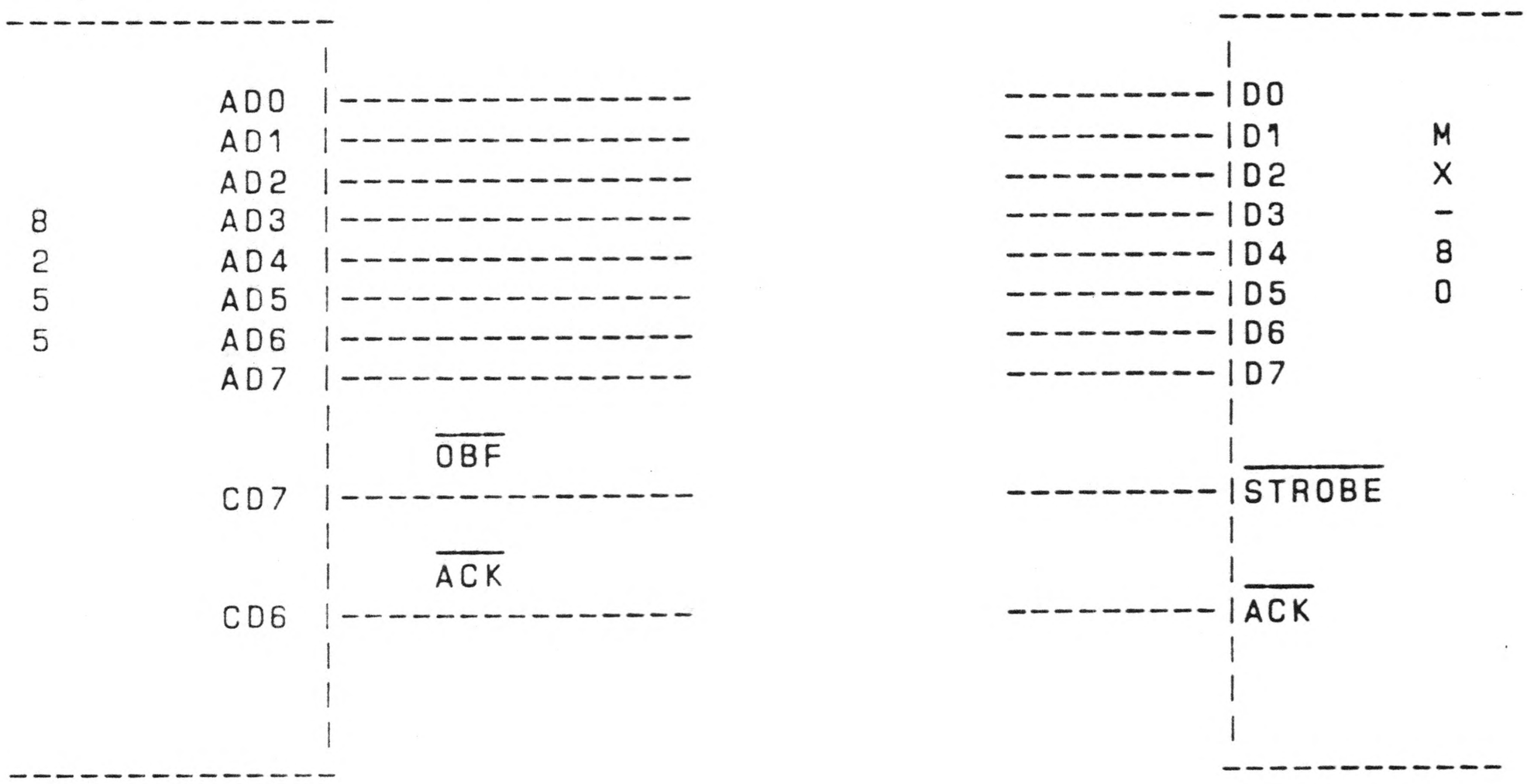
Connect Epson cable to DP-25 connector on the 'A' board and it's ready for operation.



MICRO NOTE #4

PRINTER DRIVERS FOR THE PARALLEL-EPSON MX-80

This program is used for centronics interface printers. It uses a 8255, PA0-7 are used for data transfer, OBF used for data strobe PC6 for ACK from printer.




```
LIST:
        LDA   INITFLG       ; Note, this code
        CMA                 ; needs only to be
        JZ    DATA          ; executed once
        STA   INITFLG       ;
        MVI   A, 0AFA       ;
        OUT   7             ; 8255 cmd port
                            ; (See CPU man)

                            ;MX-80 printer driver
DATA:
        MOV     A,C         ;
        OUT     04H         ;
STATLP:
        IN      06H         ;
        ANI     80H         ;
        CPI     80H         ;
        JNZ     STATLP      ;
        RET                 ;

INITFLG:
        DB      0           ;
```

MICRO NOTE #5

ANADEX 9500-SERIAL INTERFACING

The ANADEX PRINTER, MODELS DP-9500, DP-9501, DP-9000, AND DP-9001 are all similar except for maximum paper and printing width, the character dot matrix, and timing. Jumpering for serial interfacing is applicable to all models.

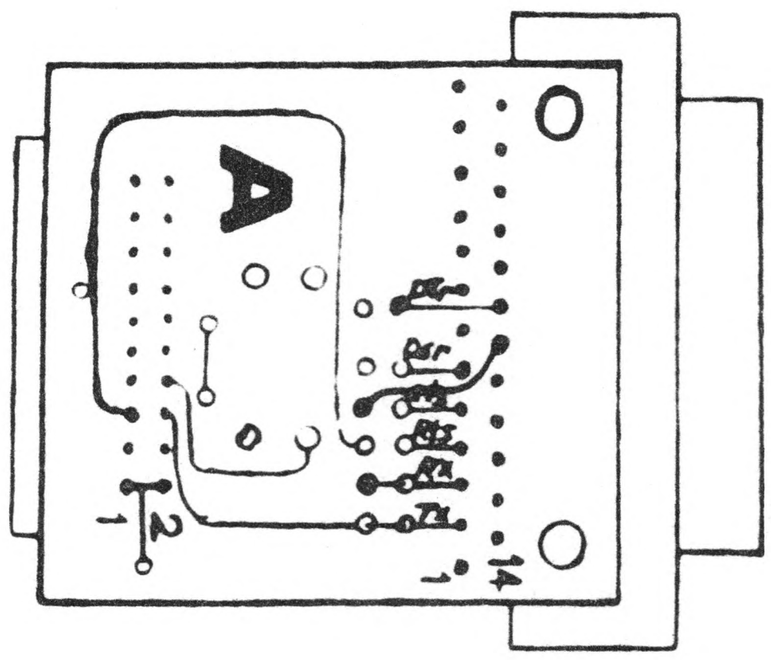
Located in the rear center of the printer are 3 DIP switches. Switches S1 ad S2 which are both 16 pin DIP switches, are customer options. See page 2 of this advisory. Switch S3, a 20 pin DIP switch should be set as follows to run with our systems:

	ON		ON	ON	ON	ON	ON			
SW3	1	2	3	4	5	6	7	8	9	10
		OFF						OFF	OFF	OFF

This will select 9600 baud, 8 bit word length, and parity disabled.

To set the baud rate for the 'A' port on the CPU board to 9600, switch top four switches identically to the bottom four switches on the 8 position DIP switch resident on the CPU board, (Port 'B' has been set at 9600 for terminal operation).

Finally, the 'A' board at the end of the serial cable of the System must have a jumper installed for handshaking to take place. Please refer to the diagram below and install a jumper from pin 19 of the DB-25 connector to CTS on the CPU side of the 'A' board. Connect a standard RS-232 cable between the 'A' board and the anadex and it will be ready for operation.



MICRO NOTE #5 (CON'T)

ANADEX MODELS DP9500, DP-9501, DP-9000, DP-9001
SWITCH SETTINGS

CONFIGURATION CONTROL SWITCH ASSIGNMENTS

S1-1	Form length 1/2 inch
S1-2	Form length 1 inch
S1-3	Form length 2 inches
S1-4	Form length 4 inches
S1-5	Form length 8 inches
S1-6	10 CPI/12 CPI (DP-9000/DP-9500); 10 CPI/16.7 CPI (DP-9001/DP-9501)
S1-7	Time Out
S1-8	£ /#
S2-1	Skip over 1/2 inch
S2-2	Skip over 1 inch
S2-3	Spare
S2-4	6/8 lines per inch
S2-5	No protocol/X=ON-X-OFF
S2-6	STX-ETX/STX-ETX-LRCC
S2-7	Wraparound/Truncate
S2-8	Auto Line Feed
S3-1	Parallel/Serial
S3-2	8/13.2 inch max. line length
S3-3	
S3-4	
S3-5	Baud Rate Selection
S3-6	
S3-7	7/8 bit word
S3-8	1/2 stop bit
S3-9	
S3-10	No parity, even parity/odd parity/echoplex
S4	EIA RS-232-C/TTY current loop
S5	Paper Out Detector
S6	Right-Hand Stop Switch
S7	Power
S8	Feed
S9	TOF Set
S10	Form Feed
S11	Self Test
S12	ON/OFF Line

MICRO NOTE #6

TEXAS INSTRUMENTS 810-SERIAL INTERFACING

Lift open the access door in front and set the auxillary control panel pencil DIP switch as follows:

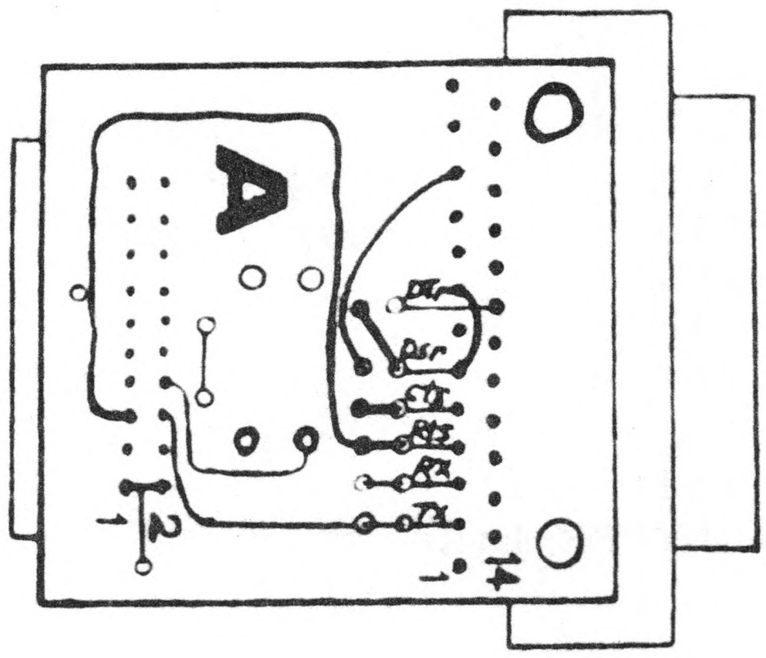
	ON					ON	ON
DIP SWITCH	1	2	3	4	5	6	7
	OFF					OFF	OFF

This will select 9600 baud, 8 bit word length, and parity disabled.

To set the baud rate for 'A' port on the CPU board to 9600, switch the top four switches identically to the bottom four switches on the 8 position DIP switch resident on the CPU board, (Port 'B' has been set at 9600 for terminal operation).

Last, the 'A' board at the end of the serial cable of the system must have jumpers installed for handshaking to take place. Please refer to the diagram below.

Note that the TX and RX lines are already jumpered on the 'A' board. The RTS and CTS lines need to be jumpered in the same manner. Run a jumper from pin 6 to pin 8 of the DB-25 connector. Also, run a jumper from pin 6 (DSR) to the open hole across from the line labeled DTR. Then run a jumper from pin 11 of the DB-25 connector to the open hole across from the line labeled DSR.



TEXAS INSTRUMENTS 810-SERIAL INTERFACING

TABLE 3-2. AUXILIARY CONTROL PANEL BAUD RATE SELECTIONS

Baud Rate		Pencil Switches		
Standard	BRO[1]	1	2	3
110	110	OFF	OFF	OFF
150	200	ON	OFF	OFF
300	300	OFF	ON	OFF
1200	1200	ON	ON	OFF
2400	2400	OFF	OFF	ON
4800	600	ON	OFF	ON
9600	9600	OFF	ON	ON
parallel[2]		ON	ON	ON

TABLE 3-3.AUXILIARY CONTROL PANEL PARITY SELECTIONS

Function	Pencil Switches 4	5
Ignore Parity	OFF	OFF
Odd Parity	ON	ON
Even Parity	ON	OFF

Table 5-1. SERIAL INTERFACE CONNECTOR SIGNALS (J13)

J13 Pin No.	Signal Name	Designation EIA	Designation C.C.I.T.T.	Source	Function
1	Protective Ground	AA	101	None	Chassis ground
2	Transmitted Data	BA	103	Printer	Held to negative EIA level in the test mode
3	Received Data	BB	104	Input Device	Received serial data
4	Request to Send	CA	105	Printer	Held to a negative EIA level
5	Clear to Send	CB	106	Input Device	(Not Used)
6	Data Set Ready	CC	107	Input Device	Must be at positive EIA level for the printer to receive data
7	Signal Ground	AB	102	None	Return path for data and control signals
8	Carrier Detect	CF	109	Input Device	Must be at positive EIA level for the printer to receive data
9	+12 Volts	—		Printer	May be used as bias voltage for inputs to printer (1000-ohms source impedance)
10	12 Volts	—	—	Printer	May be used as bias voltage for inputs to printer. (1000-ohms source impedance)
11	Reverse Channel	SCA	120	Printer	Held to negative EIA level when *standard* printer is busy, and to positive EIA level when *standard* is not busy; these levels are inverted in IRC printer
20	Data Terminal Ready	CD	108.2	Printer	Held to positive EIA level when *standard* printer is on line or when the DNB printer is on line and not busy; and to negative EIA level when *standard* printer is off line, or when DNB printer is off line or busy

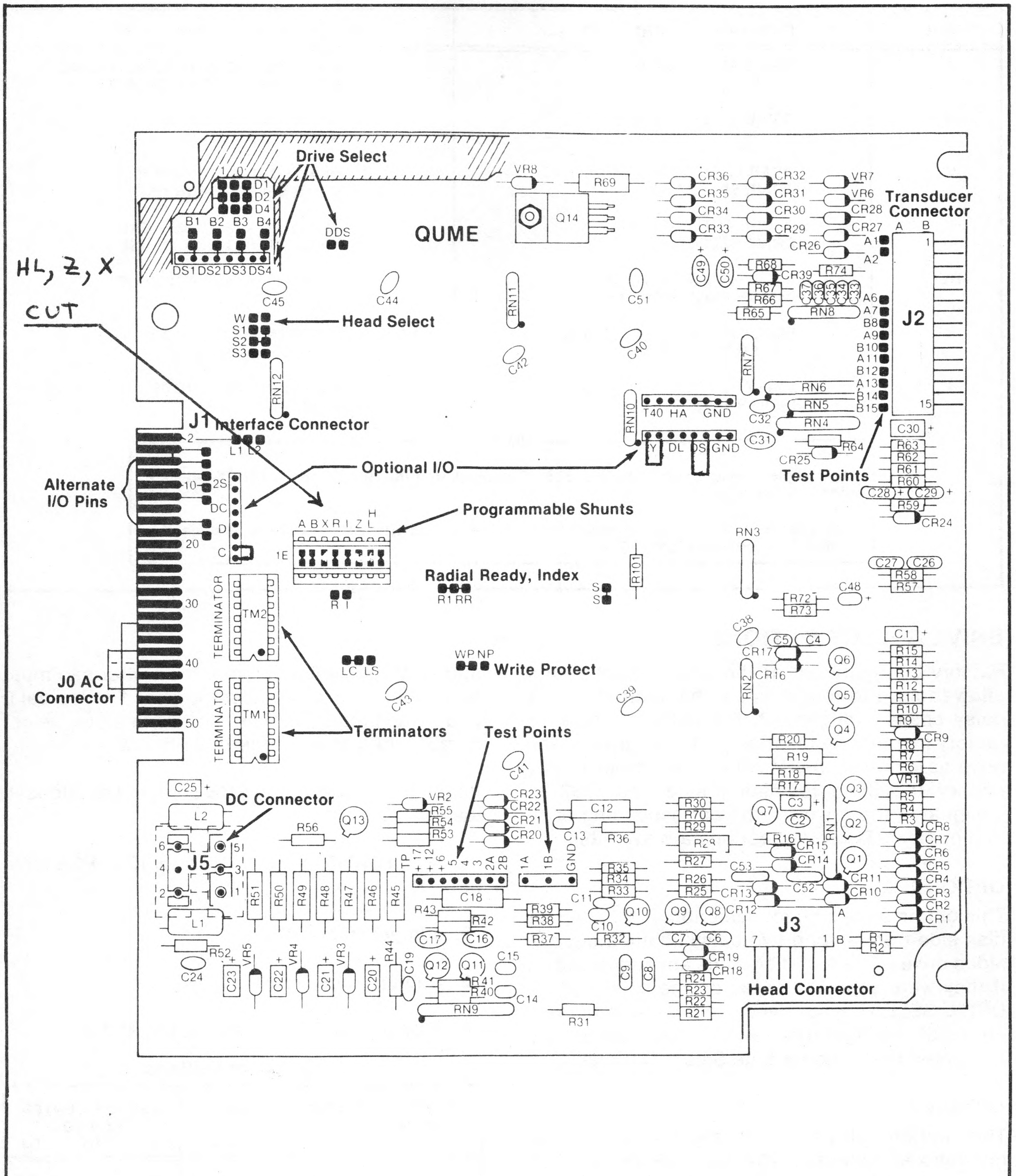
QUME DT-8

CUSTOMER STRAPPABLE OPTIONS

The QumeTrak 842 can be modified by the user to alter functions that were described previously. These modifications are made by adding or deleting traces and by using the Alternate I/O pins. The following table shows the configuration of the option traces and jumpers as shipped from the factory.

Factory Configuration of the Option Traces

DESIGNATOR	DESCRIPTION	FROM FACTORY	
		OPEN	SHORT
DS1 - DS4	**DRIVE SELECT ADDRESS** PINS (UP TO 4 DRIVES)	**DS2, DS3 DS4**	**DS1**
A,B,X	RADIAL **HEAD LOAD**		X
Z	**IN USE** FROM **DRIVE SELECT**		X
HL	STEPPER POWER FROM **HEAD LOAD**		X
R	ALTERNATE OUTPUT **READY** PAD		X
I	ALTERNATE OUTPUT **INDEX** PAD		X
C	ALTERNATE INPUT **HEAD LOAD**	X	
D	ALTERNATE INPUT **IN USE**	X	
DC	ALTERNATE OUTPUT **DISK CHANGE**	X	
2S	ALTERNATE OUTPUT **2-SIDED DISK**	X	
DS	STEPPER POWER FROM **DRIVE SELECT**	X	
Y	**IN USE** FROM **HEAD LOAD**	X	
DL	DOOR LOCK LATCH	X	
RR	RADIAL READY		X
RI	RADIAL INDEX		X
WP	INHIBIT WRITE WHEN **WRITE PROTECT**		X
NP	ALLOW WRITE WHEN **WRITE PROTECT**	X	
D1,D2,D4,DDS	DRIVE ADDRESS SELECT (UP TO 8 DRIVES)	X	
B1 - B4	TWO-SIDED DRIVE SELECT	X	
S1 - S3	HEAD SELECT OPTION	**S1,S3**	S2
T40	TEST TRACK 40	X	
HA	TEST ACTUATE HEAD LOAD	X	
4,6,8,10, 12,16,18,24	ALTERNATE I/O PINS	X	



PCB Jumper Locations

PROGRAMMABLE SHUNT

A 16-pin programmable shunt is provided for the seven commonly used tracecut options. All of these traces are usually shorted when shipped from the factory. The shorted and open functions of the pluggable options are shown in the following table.

Programmable Shunt Trace Functions

TRACE	FUNCTION - SHORTED (STD.)	FUNCTION - OPEN (OPTION)
A	**DRIVE SELECT** ACTIVE	**DRIVE SELECT** ACTIVE, **HEAD LOAD** ACTIVE FROM **DRIVE SELECT**
B	**DRIVE SELECT** ACTIVE	**DRIVE SELECT** ACTIVE **HEAD LOAD** ACTIVE
X	**HEAD LOAD** ACTIVE FROM **DRIVE SELECT**	**HEAD LOAD** ACTIVE FROM **DRIVE SELECT** OR **IN USE**
Z	**IN USE** ACTIVE FROM **DRIVE SELECT**	**IN USE** ACTIVE FROM **HEAD LOAD** OR **OPTIONAL IN USE** I/O
HL	STEPPER MOTOR POWER ACTIVE FROM **HEAD LOAD**	STEPPER MOTOR POWER IS ALWAYS ACTIVE
R	**READY** IN MULTIPLEX MODE (REFER TO NOTE 1)	**READY** IN RADIAL MODE (REFER TO NOTE 2)
I	**INDEX** IN MULTIPLEX MODE (REFER TO NOTE 1)	**INDEX** IN RADIAL MODE (REFER TO NOTE 2)

NOTES:

1. Multiplex mode requires DRIVE SELECT to check that the READY and/or INDEX lines are active on any drive.

2. Radial mode allows the controller to check that the READY and/or INDEX lines are active on any drive without DRIVE SELECT.

DRIVE SELECT OPTIONS

Factory configuration of the Drive Select circuit allows for up to four drives to be addressed in a daisy chain arrangement. Normally, a jumper is factory installed in location DS1, designating the drive as number one. Moving this jumper to any one of the three remaining locations (DS2-4), designates the drive for the corresponding position number. Two other options are available.

OPTION 1

This option allows direct selection of a particular disk side from among two daisy chained double-sided drives. The four disk sides are addressed as if they were separate drives on the existing four DRIVE SELECT input lines. Detailed instructions on input configuration and jumper status are presented below under Side Select Options.

OPTION 2

This option allows up to eight drives to be multiplexed together. The four DRIVE SELECT lines are used for addressing the drive. DRIVE SELECT 1 is used to enable the DRIVE SELECT 2 (binary 1), DRIVE SELECT 3 (binary 2), and DRIVE SELECT 4 (binary 4) address lines. The logical drive assignment is accomplished by properly jumpering traces D1, D2, and D4, where D1 = (binary 1), D2 = (binary 2), D4 = (binary 4). The jumper pads are arranged to appear as a 3 X 3 matrix. Within each column, the center pad must be jumpered to an adjacent logical 1 or logical 0 pad. The proper jumpers for each drive select number are shown in the table below.

Installation (refer to the PCB Jumper Locations illustration):

1. Add a 74L85, 4-bit comparator IC in PCB location 1K.

2. Jumper trace DDS.

3. Unplug trace DS1-DS4.

4. Properly jumper traces D1, D2, and D4.

Drive Select Options

DRIVE NUMBER	DRIVE SELECT INPUT				JUMPER CENTER PAD TO:		
	1	2	3	4	D1	D2	D4
0	L	H	H	H	0	0	0
1	L	L	H	H	1	0	0
2	L	H	L	H	0	1	0
3	L	L	L	H	1	1	0
4	L	H	H	L	0	0	1
5	L	L	H	L	1	0	1
6	L	H	L	L	0	1	1
7	L	L	L	L	1	1	1

L = LOW LEVEL, H = HIGH LEVEL

HEAD LOAD OPTIONS

When a factory configured drive is selected, its heads are loaded and the stepper motor is energized. Three other options are available.

OPTION 1

This option allows a drive to be selected without loading the heads or enabling the stepper motor. The advantage is that the output status signals can be monitored while the head is unloaded, thereby extending the media life.

Installation (refer to the PCB Jumper Locations illustration):

1. Open trace X on the programmable shunt with a small screwdriver.

2. Plug trace C on the optional I/O pins near connector J1.

OPTION 2

This option allows a drive to be selected and the stepper motor to be enabled without loading the heads. As an example of this option, initial reset to Track 00 at power-up can be performed without the READY condition.

Installation (refer to the PCB Jumper Locations illustration):

1. Open traces B and HL on the programmable shunt with a small screwdriver.

2. Plug trace C on the optional I/O pins near connector J1.

3. Plug trace DS on the optional I/O pins near connector J2.

OPTION 3

This option (RADIAL READY) allows a drive to load the heads without selecting the drive or enabling the stepper motor. The advantage is that the heads can be kept loaded on all the drives, thereby eliminating the head load time when the drive is performing a disk copy operation.

Installation (refer to the PCB Jumper Locations illustration):

1. Open trace A and HL on the programmable shunt with a small screwdriver.

2. Plug trace C on the optional I/O pins near connector J1.

3. Plug trace DS on the optional I/O pins near connector J2.

SIDE SELECT OPTIONS

Normally, a R/W head is selected by the SIDE SELECT interface line in a daisy chain system of up to four drives. Two options are available.

OPTION 1 (Side Select From Direction Line)

This option allows both SIDE SELECT and DIRECTION to be multiplexed on the same DIRECTION line. DIRECTION can be used in this way because it is not being used during reading or writing (STEP must accompany DIRECTION to initiate head movement).

Installation (refer to the PCB Jumper Locations illustration):

Cut trace S2 and jumper trace S1.

OPTION 2 (Side Select From Drive Select Lines)

This option uses the existing DRIVE SELECT lines to address up to two double-sided drives. It selects up to four sides as if the sides were separate drives.

Installation (refer to the PCB Jumper Locations illustration):

Cut trace S2 and jumper trace S3. Disk sides are then selected according to the table below.

Side Select Options

DRIVE NUMBER	HEAD SELECT	DRIVE SELECT INPUT 1	2	3	4	TRACES
1	0	L	H	H	H	PLUG DS1
1	1	H	L	H	H	JUMPER B2
2	0	H	H	L	H	PLUG DS3
2	1	H	H	H	L	JUMPER B4

L = LOW LEVEL, H = HIGH LEVEL

RADIAL READY OPTION

Normally, the READY line from a factory configured drive is only available to the interface when it is selected. This option enables the user to monitor the READY line of each drive on the interface continuously.

Installation (refer to the PCB Jumper Locations illustration):

1. Cut trace RR (near the center of the PCB).

2. Open trace R on the programmable shunt with a small screwdriver.

3. Jumper pad R (directly below the programmable shunt) to one of the available Alternate I/O pins (4, 6, 8, or 24). READY will then appear on the chosen Alternate I/O pin.

RADIAL INDEX OPTION

Normally, the INDEX line from a factory configured drive is only available to the interface when the drive is selected. This option enables the user to monitor the INDEX line of each drive on the interface continuously.

Installation (refer to the PCB Jumper Locations illustration):

1. Cut trace R1 (near the center of the PCB).

2. Open trace R on the programmable shunt with a small screwdriver.

3. Jumper pad (directly below the programmable shunt) to one of the available Alternate I/O pins (4, 6, 8, or 24). READY will then appear on the chosen Alternate I/O pin.

IN USE OPTIONS

Normally, the In Use Led indicator will be energized when DRIVE SELECT is active. The door solenoid will be activated when DRIVE SELECT and READY are active. The three options listed below will energize the IN Use LED when activated to a low level.

OPTION 1

The option will energize the In Use LED when the DRIVE SELECT or IN USE line is active.

Installation (refer to the PCB Jumper Locations illustration):

Plug trace D on the optional I/O pins near connector J1.

OPTION 2

This option will energize the In Use LED when the HEAD LOAD or IN USE line is active.

Installation (refer to the PCB Jumper Locations illustration):

1. Open trace Z on the programmable shunt with a small screwdriver.

2. Plug trace D on the optional I/O pins near connector J1.

3. Plug trace Y on the optional I/O pins near connector J2.

OPTION 3

This option will energize the In Use LED only when the IN USE line is active.

Installation (refer to the PCB Jumper Locations illustration):

1. Open trace Z on the programmable shunt with a small screwdriver.

2. Plug trace D on the optional I/O pins near connector J1.

DOOR LOCK LATCH OPTION

With this option, the door lock actuator can be latched without maintaining the IN USE signal throughout the door lock interval. IN USE may be activated by DRIVE SELECT.

Installation (refer to the PCB Jumper Locations illustration):

1. Plug trace D on the optional I/O near connector J1.

2. Plug trace DL on the optional I/O pins near connector J2.

WRITE PROTECT OPTION

With this option installed, a Write Protected disk will not inhibit writing but it will be reported to the controller.

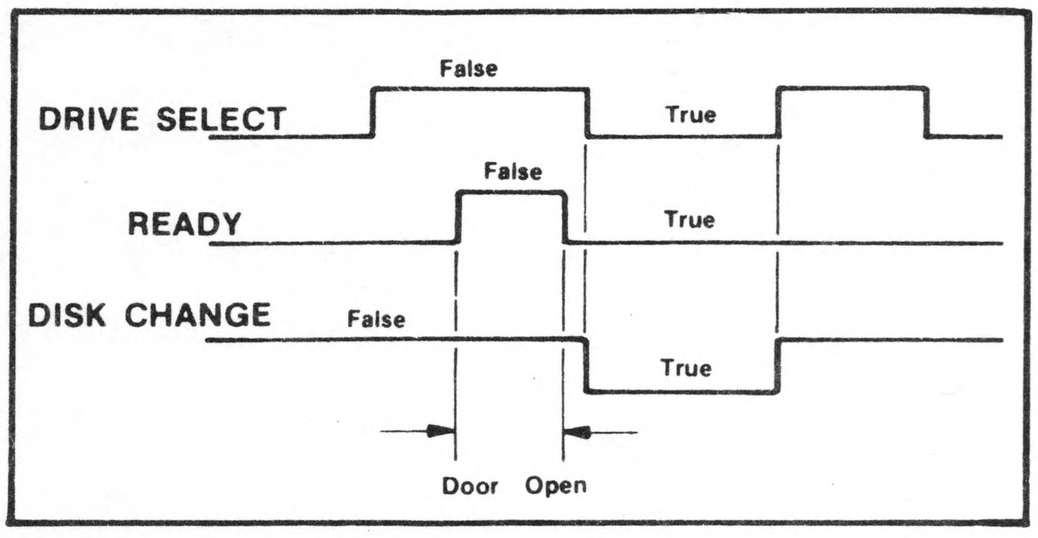
Installation (refer to the PCB Jumper Locations illustration):

Cut trace WP and jumper trace NP (located near the center of the PCB).

DISK CHANGE (ALTERNATE OUTPUT)

An active low level on this option line indicates that the READY signal has gone false (door opened) after DRIVE SELECT went false. The DISK CHANGE circuit is reset on the true to false (low/high) transition of DRIVE SELECT provided that the drive is READY. Refer to the following illustration.
Installation (refer to the PCB Jumper Locations illustration):

Plug trace DC on the optional I/O pins near connector J1.



Disk Change Timing

TWO-SIDED (ALTERNATE OUTPUT)

A low level on this line indicates that a two-sided disk is in use, and a high level indicates that a single-sided disk is in use.

Installation (refer to the PCB Jumper Locations illustration):

Plug trace 2S on the optional I/O pins near connector J1.

XOR